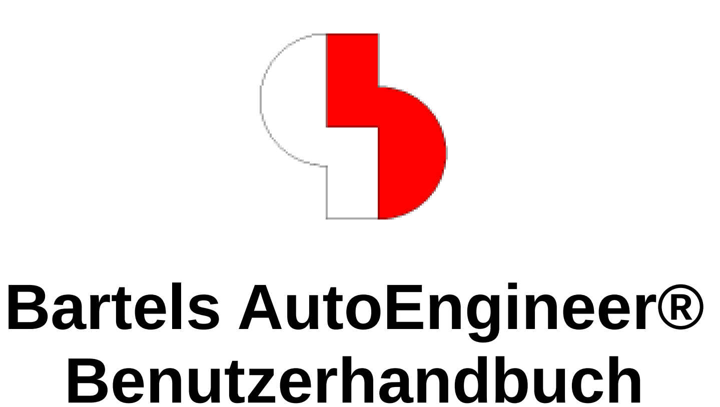


Bartels AutoEngineer®
Benutzerhandbuch

Bartels AutoEngineer Benutzerhandbuch
Herausgeber: Bartels System GmbH, München
Stand: November 2013

Vorwort

Das Bartels AutoEngineer® - Benutzerhandbuch enthält die Bedienungsanleitung für das **Bartels AutoEngineer**-CAE/CAD/CAM-System. Im Bartels AutoEngineer® - Benutzerhandbuch finden Sie detaillierte Informationen zu den folgenden Themenschwerpunkten:

- Einleitung: Systemarchitektur, allgemeine Bedienungshinweise, Designdatenbank
- Schaltungsentwurf (CAE), **Schematic Editor**
- Netzlistenverarbeitung, Forward- und Backward-Annotation
- Leiterplattenentwurf (CAD) und Fertigungsdatenerzeugung (CAM), **Layouteditor** für grafisch-interaktives Leiterkartenlayout, **Autoplacement**, Flächenautomatik, Autorouting, **CAM-Prozessor**, **CAM-View**
- IC-/ASIC-Design, **Chipeditor** für grafisch-interaktives IC-Maskenlayout, **Cellplacer** und **Cellrouter** für "Place & Route", Import- und Export von GDS- und CIF-Daten
- Neuronales Regelsystem
- Utilityprogramme

Der Leser sollte vertraut sein mit der Benutzung seines Betriebssystems und mit der Handhabung eines auf seinem System verfügbaren Editors zur Erstellung von ASCII-Dateien.

Beachten Sie bitte vor Verwendung der in dieser Dokumentation enthaltenen Informationen und der darin beschrieben Produkte die Copyright-Hinweise. Der Leser sollte darüber hinaus auch mit den in dieser Dokumentation verwendeten Begriffen und Konventionen vertraut sein.

Gliederung

Kapitel 1 beschreibt die Systemarchitektur des **Bartels AutoEngineer** und gibt grundsätzliche Hinweise zu dessen Bedienung und zur Designdatenbank.

Kapitel 2 erläutert die Handhabung des Programm-Moduls **Schematic Editor** für den Entwurf von Schaltplänen.

Kapitel 3 beschreibt die Arbeitsweise von **Packager** und **Backannotation**.

Kapitel 4 beschreibt den **Layouteditor**, die **Autoplacement**-Funktionen und den **Autorouter** für den Entwurf des Leiterkartenlayouts und erläutert die Handhabung des **CAM-Prozessors** und des **CAM-View**-Moduls zur Erzeugung und Bearbeitung der Fertigungsdaten für die Leiterkartenproduktion.

Kapitel 5 erläutert die Handhabung der Programm-Module **Chip Editor**, **Cell Placer** und **Cellrouter** für das grafisch-interaktive und automatisierte Design von IC-Maskenlayouts.

Kapitel 6 beschreibt das im **Bartels AutoEngineer** integrierte **Neuronale Regelsystem**, d.h. die Bartels Rule Specification Language zur Definition neuronaler Regeln, die Handhabung des Bartels Rule System Compilers zur Kompilierung von Regeldefinitionen, sowie die Möglichkeiten der Anwendung von neuronalen Regeln im BAE-Designprozess.

Kapitel 7 beschreibt die Utilityprogramme des **Bartels AutoEngineer**.

Weitere Dokumentation

Die Bartels AutoEngineer® - Installationsanleitung beschreibt die Konfigurationen und Systemvoraussetzungen des **Bartels AutoEngineer** und enthält detaillierte Anweisungen zur Installation des **Bartels AutoEngineer** auf unterschiedlichen Hardware- und Softwareplattformen.

Die Bartels AutoEngineer® - Symbol- und Bauteilbibliotheken Dokumentation enthält detaillierte Informationen zu den mit dem **Bartels AutoEngineer**-CAE/CAD/CAM-System ausgelieferten Symbol- und Bauteilbibliotheken.

Das Bartels User Language - Programmierhandbuch enthält die Beschreibung der **Bartels User Language**-Programmiersprache sowie ausführliche Informationen über die Art der Einbindung sowie die Möglichkeiten der Anwendung im **Bartels AutoEngineer**-EDA-System. Im Bartels User Language - Programmierhandbuch finden Sie detaillierte Informationen zu den folgenden Themenschwerpunkten:

Grundkonzept und Sprachbeschreibung der **Bartels User Language**

- das Bartels User Language-Programmiersystem: User Language Compiler und User Language Interpreter
- User Language-Beispielprogramme; Informationen zu den mit dem Bartels AutoEngineer ausgelieferten User Language-Programmen
- Beschreibung der Datentypen für den Zugriff auf die Designdaten im Bartels AutoEngineer
- Beschreibung der in der Bartels User Language implementierten Systemfunktionen

Wünsche, Anregungen, Fragen, Probleme

Für Hinweise auf Fehler sowie Wünsche und Anregungen in Bezug auf die Implementierung neuer oder die Weiterentwicklung bestehender Funktionen bzw. Programmteile des **Bartels AutoEngineer** sind wir Ihnen dankbar. Sollten Sie Fragen zum **Bartels AutoEngineer** haben, oder Probleme bei seiner Benutzung auftreten, so wenden Sie sich bitte an unsere Support-Abteilung. Unsere Anschrift können Sie der Bartels Website unter http://www.bartels.de entnehmen.

Begriffe

Der Leser sollte vertraut sein mit den folgenden in der **Bartels AutoEngineer**-Dokumentation immer wiederkehrenden Begriffen:

Maus	Zeigegerät (Maus, Trackball, etc.) zum Bewegen des Menübalkens und des Fadenkreuzes sowie zur Aktivierung von Funktionen verwendet
Info-Feld	Feld zur Anzeige von System-Statusmeldungen rechts oben am Bildschirm
Hauptmenü	Fest vorgegebene Funktionsauswahl im oberen Bereich der rechten Bildschirmseite zur Selektion eines Menüs
Menü	Über Hauptmenü eingestellte Funktionsauswahl im unteren Bereich der rechten Bildschirmseite
Untermenü	Aus Menüfunktion aufgerufene, untergeordnete Funktionsauswahl im unteren Bereich der rechten Bildschirmseite
Grafikarbeitsbereich	Arbeitsbereich für Grafik-Interaktionen im linken, oberen Bildschirmbereich
Status-Zeile	Unterste Bildschirmzeile zur Anzeige von System-Statusmeldungen bzw. für Benutzerabfragen
Menübalken	Menükursor zum Anwählen einer Menüfunktion
Fadenkreuz	Kursor im Grafikarbeitsbereich
Menüprompt	Benutzerabfrage in der Status-Zeile
Popupmenü	Optional über dem Grafikarbeitsbereich angezeigtes Auswahlmenü für funktionsspezifische Objekte, Elemente oder Arbeitsschritte
Button	Selektierbarer Popupmenüeintrag zur Anwahl eines speziellen Elements oder zur Aktivierung einer menüspezifischen Funktion
Funktion anwählen	Den Menübalken mittels Maus auf die Menüfunktion positionieren
Aktivieren	Betätigen der Maustaste
Pick	Ein zu manipulierendes Element mit dem Grafikkursor selektieren
Positionieren (auch: **Place**)	Ein Element mittels Grafikkursor im Grafikarbeitsbereich positionieren
Selektieren	Auswählen eines zu bearbeitenden Elements oder einer Funktion durch Betätigen der Maustaste
Bestätigen	Die Ausführung einer über Benutzerabfrage verifizierten Funktion veranlassen

In der **Bartels AutoEngineer**-Dokumentation werden die folgenden Akronyme verwendet:

BAE	Akronym zur Identifikation der **Bartels AutoEngineer**-EDA-Software
BAEICD	Akronym für das **Bartels AutoEngineer**-IC/ASIC-Designsystem, welches optional in workstationbasierenden BAE-Konfigurationen enthalten ist
SCM	Akronym für das **Schematic Editor**-Programm-Modul für den Schaltungsentwurf und zur Stromlaufplanerfassung im **Bartels AutoEngineer**
GED	Akronym für den grafischen **Layouteditor** (**Layouteditor**) des **Bartels AutoEngineer** PCB-Designsystems
AR	Akronym für den **Autorouter** des **Bartels AutoEngineer** PCB-Designsystems
NAR	Akronym für den **Neuronalen Autorouter** des **Bartels AutoEngineer** PCB-Designsystems
CAM	Akronym für den **CAM-Prozessor** des **Bartels AutoEngineer** PCB-Designsystems
CV	Akronym für das **CAM-View**-Programm-Modul des **Bartels AutoEngineer** PCB-Designsystems
CED	Akronym für das **Chip Editor**-Programm-Modul des **Bartels AutoEngineer** IC/ASIC-Designsystems
CP	Akronym für das **IC-Autoplacement**-Programm-Modul des **Bartels AutoEngineer** IC/ASIC-Designsystems
CR	Akronym für das **IC-Autorouter**-Programm-Modul des **Bartels AutoEngineer** IC/ASIC-Designsystems
UL	Akronym für die **Bartels User Language**-Programmiersprache
ULC	Akronym für den **Bartels User Language Compiler**
ULI	Akronym für den **Bartels User Language Interpreter**

Konventionen

Soweit nicht anders vermerkt sind in der **Bartels AutoEngineer**-Dokumentation folgende symbolische Konventionen relevant:

`Lineprint`	Schreibmaschinenzeichensatz kennzeichnet durch das System ausgegebenen Text.
Boldface	Fett gedruckte Worte oder Zeichen in Format- oder Kommandobeschreibungen kennzeichnen feststehende Begriffe oder syntaktische Terminalzeichensequenzen, also direkt einzusetzende Kommando- oder Schlüsselwörter.
Emphasize	Emphatische Textauszeichnung dient der optischen Hervorhebung.
" "	Anführungszeichen dienen der Kennzeichnung von (Pfad-)Namen oder spezifizieren direkt einzugebende Zeichen(sequenzen).
[ ]	Eckige Klammern in Format- oder Kommandobeschreibungen umschließen wahlweise angebbare Elemente.
{ }	Geschweifte Klammern in Format- oder Kommandobeschreibungen umschließen eine Liste von Elementen, aus denen eines anzugeben ist.
|	Ein vertikaler Strich trennt Elemente aus einer Liste wahlweise angebbarer Elemente.
< >	Gewinkelte Klammern umschließen den logischen Namen einer (zu betätigenden) Taste oder eine semantisch zu ersetzende syntaktische Variable in einer Format- oder Kommandobeschreibung.
>	Fett gedruckte Größerzeichen innerhalb Schreibmaschinenzeichensatz kennzeichnen Eingabeaufforderungen auf Betriebssystemebene.
...	Horizontale Auslassungspunkte kennzeichnen die wahlweise Wiederholbarkeit des vorhergehenden Elements in einer Format- oder Kommandobeschreibung oder die Auslassung irrelevanter Teile eines Beispieles oder einer Abbildung.
:	Vertikale Auslassungspunkte kennzeichnen die Auslassung irrelevanter Teile einer Abbildung, eines Beispieles, oder einer Format- oder Kommandobeschreibung.
	beliebige Maustaste (MB)
	Linke Maustaste (LMB)
	Mittlere Maustaste (MMB)
	Rechte Maustaste (RMB)
↵	Tastatur(eingabe) - Return-/Eingabetaste (CR)
a **b** ...	Tastatur(eingabe) - Standardtaste(n)
F1 **F2** ...	Tastatur(eingabe) - Funktionstaste(n)
`filename`	Datei- bzw. Verzeichnispfadname.
`keyword`	Syntaktische Terminalzeichensequenz, also direkt einzusetzendes Kommando bzw. Schlüsselwort.
`message`	Von BAE oder System angezeigte Status- oder Fehlermeldung.
Menu	**Bartels AutoEngineer** Menü.
Menu Function	**Bartels AutoEngineer** Menüfunktion.
Menu Option	**Bartels AutoEngineer** Menüoption.
Button	**Bartels AutoEngineer** (Popup-)Menübutton.
ul.ulc	(Hypertextlink zu) **Bartels User Language**-Programmquellcodedatei.
ul.ulh	(Hypertextlink zu) **Bartels User Language**-Includedatei.

ULPROG	(Hypertextlink zu) **Bartels User Language**-Programmbeschreibung.
ul_function	(Hypertextlink zu) **Bartels User Language**-Systemfunktionsbeschreibung.
UL_INDEX	(Hypertextlink zu) **Bartels User Language**-Indextypbeschreibung.
UTILPROG	(Hypertextlink zu) **Bartels AutoEngineer** Utilityprogrammbeschreibung.
neue Funktion	Neue Funktionen die mit regulären wöchentlichen Updates/Builds verfügbar gemacht werden, werden in der Onlinedokumentation hervorgehoben.

In Beispielen für höhere Programmiersprachen und Interpretationssprachen, in Objektbeschreibungen mittels Spezifikationssprachen, bei der Verwendung in Syntaxbeschreibungssprachen, etc. erhalten obige Sonderzeichensequenzen wieder die in der entsprechenden Sprache festgelegte Bedeutung.

Inhaltsverzeichnis

Tabellen

Abbildungen

Kapitel 1
Einleitung

Dieses Kapitel enthält einleitende Anmerkungen zum **Bartels AutoEngineer**, beschreibt die Systemarchitektur und gibt grundsätzliche Hinweise zu dessen Bedienung und zur Designdatenbank.

Inhalt

Abbildungen

1.1 Produktinformationen

Bartels AutoEngineer (BAE) ist ein integriertes EDA-Softwarepaket mit mächtigen CAE-, CAD- und CAM-Programmmoduln für Schaltungsentwurf, Leiterkartenlayout und IC-/ASIC-Design. Das System stellt die konsequente Weiterentwicklung des **Bartels AutoEngineer** dar, der weltweit von großen CAD-Häusern unter verschiedenen Namen angeboten wird und aufgrund seiner Leistungsfähigkeit schnell zu einem de-facto-Industriestandard geworden ist.

1.1.1 Softwarekonfigurationen

Die **Bartels AutoEngineer** Software wird in den folgenden Ausbaustufen bzw. Konfigurationen angeboten:

- **Bartels AutoEngineer Schematics**
- **Bartels AutoEngineer Light**
- **Bartels AutoEngineer Economy**
- **Bartels AutoEngineer Professional**
- **Bartels AutoEngineer HighEnd**
- **Bartels AutoEngineer IC Design**
- **Bartels AutoEngineer FabView**

Alle Softwarekonfigurationen des **Bartels AutoEngineer** werden mit einheitlichen Benutzeroberflächen in unterschiedlichen Landessprachen (deutsch, englisch, etc.) angeboten. Die mit dem **Bartels AutoEngineer** erstellten Designdaten können binärkompatibel zwischen den unterstützten Hardware- bzw. Betriebssystemplattformen (Windows, Linux/Unix, DOS, usw.) ausgetauscht werden.

Bartels AutoEngineer Professional

Das System **Bartels AutoEngineer Professional** inklusive Schaltplan- und Leiterkarten-Layout-Modul ist das Basissystem der BAE-Software und entspricht dem in diesem Handbuch beschriebenen Softwareprodukt. **BAE Professional** ist auf PC-Systemen unter Windows, Linux und DOS ablauffähig. In der **BAE Professional** Software sind die folgenden Komponenten enthalten:

- **Schaltplaneditor** mit hierarchischem Design
- Forward- und Backward-Annotation
- Leiterkarten-Layoutsystem mit **Layouteditor**, **Autoplacement**, Flächenautomatik, **Bartels AutoEngineer**, **Bartels Autorouter®**
- **CAM-Prozessor** und **CAM-View** mit Gerber-Viewer
- integriertes, objektorientiertes Datenbanksystem (DDB, Design DataBase)
- integriertes **Neuronales Regelsystem**
- **Bartels User Language Compiler**, **User Language**-Programme mit Quellcode
- Utilityprogramme zur Bibliotheksverwaltung, zum Importieren von Fremdnetzlisten, usw.
- umfangreiche Bauteilbibliotheken für Stromlauf und Layout
- Beispiel-Jobs
- Dokumentation (Bartels AutoEngineer Benutzerhandbuch, Bartels User Language Programmierhandbuch)

Um speziellen Einsatzgebieten gerecht zu werden, besteht auch die Möglichkeit, das *frei verfügbare* Schaltplanpaket **BAE Schematics** des **BAE Professional** als alleinstehendes Softwarepaket zu betreiben. Zu Test- und Evaluierungszwecken sind Demo-Softwarekonfigurationen des **Bartels AutoEngineer Professional** frei erhältlich (**BAE Demo**; volle **BAE Professional** Funktionalität mit Ausnahme der Datenausgabe).

Bartels AutoEngineer Light

Das extrem preisgünstige Einstiegspaket **Bartels AutoEngineer Light** ist für Schulungszwecke bzw. für semi-professionelle Anwender gedacht. **BAE Light** ist auf PC-Systemen unter Windows, Linux und DOS ablauffähig. **BAE Light** bietet dieselbe Funktionalität wie **BAE Professional**, allerdings mit den folgenden Einschränkungen:

- maximale Layout- bzw. Leiterkartengröße limitiert auf 180mm*120mm
- maximal 2 Signallagen für die manuelle Bearbeitung und die Entflechtung im **Autorouter**
- keine Versorgungslagen
- Layouts aus **BAE Professional**, **BAE Economy** und **BAE HighEnd** nur lad- und weiterverarbeitbar, wenn die Einschränkungen von **BAE Light** eingehalten wurden

Bartels AutoEngineer Economy

Das preisgünstige Einstiegspaket **Bartels AutoEngineer Economy** (auch bekannt unter der alten Bezeichnung **Bartels AutoEngineer Educate/Entry**) ist für Schulungszwecke bzw. für professionelle Anwender mit limitierten Anforderungen gedacht. **BAE Economy** ist auf PC-Systemen unter Windows, Linux und DOS ablauffähig. **BAE Economy** bietet dieselbe Funktionalität wie **BAE Professional**, allerdings mit den folgenden Einschränkungen:

- maximale Layout- bzw. Leiterkartengröße limitiert auf 350mm*200mm
- maximal 4 Signallagen simultan durch den **Autorouter** entflechtbar (jedoch wie in **BAE Professional** 100 manuell bearbeitbare Signallagen sowie bis zu 12 Versorgungslagen sowohl für das manuelle Routing als auch die Entflechtung im **Autorouter**)
- Layouts aus **BAE Professional** und **BAE HighEnd** nur lad- und weiterverarbeitbar, wenn die Einschränkungen von **BAE Economy** eingehalten wurden

Bartels AutoEngineer HighEnd

Das System **Bartels AutoEngineer HighEnd** ist sowohl auf Workstations als auch auf Windows- und Linux-PC-Plattformen verfügbar. **BAE HighEnd** nutzt spezielle Eigenschaften dieser Betriebssysteme (Multitasking, Multiwindowing, virtuelle Speicherverwaltung, usw.) zur Bereitstellung mächtiger zusätzlicher Funktionen und Leistungsmerkmale wie etwa:

- HighSpeed Kernel
- Kommunikation zwischen unterschiedlichen BAE-Modulen über integriertes Message-System
- integriertes Multitasking zur simultanen Anzeige unterschiedlicher Projektansichten
- globales Netz-Highlight, Cross-Probing
- Platzierung von Layoutbauteilen nach Schaltplan
- selektive Kurzschlussanzeige im Layoutsystem
- extrem schnelle Mincon- bzw. Airlineberechnung
- DRC mit Multiprozessorunterstützung
- Lagenaufbau für impedanzrichtige Leitungen spezifizierbar
- lagenspezifische Mindestabstände für DRC und Flächenfüllautomatik
- interne Datenstrukturen optimiert für schnelles Autorouting
- neuronales Regelsystem mit erweiterten Funktionen
- regelgesteuerter **Neuronaler Autorouter**
- netztypspezifische Routingbereiche
- netzspezifische maximale Viaanzahl
- netzspezifische maximale Verbindungslänge
- Unterdrückung nicht angeschlossener Innenlagenpads in der CAM-Ausgabe

BAE HighEnd ist in beiden Richtungen datenkompatibel zu **BAE Professional**; erforderliche Datenumsetzungen werden beim Laden automatisch durchgeführt.

Bartels AutoEngineer IC Design

BAE HighEnd kann wahlweise zu einem kompletten IC- und ASIC-Designsystem aufgerüstet werden. **Bartels AutoEngineer IC Design (BAEICD)** ist ein durchgängiges CAD/CAM-System für den Entwurf von integrierten Schaltkreisen (Gate Arrays, Standardzellen, Custom-ICs bzw. ASICs). **BAEICD** besteht aus den Komponenten **IC-Maskeneditor**, **IC-Autoplacement**, **IC-Autorouter** und **IC-DRC** (Design Rule Check). Standardschnittstellen zu GDS-II und CIF für die Übernahme von Fremddaten bzw. die Ausgabe der Fertigungsdaten (Maskendaten, Bondingdaten, etc.) mit einem Modul zur visuellen Prüfung von CIF-Daten sind ebenfalls integriert. Die Übernahme der Netzlistendaten erfolgt üblicherweise kontrolliert über den BAE **Packager** nach Erstellung des Stromlaufplans mit dem BAE-Schaltplanmodul, welches selbstverständlich das für den Entwurf integrierter Schaltkreise zwingend notwendige Leistungsmerkmal des hierarchischen Schaltplandesigns aufweist. Alternative Lösungen zur Übernahme von Fremdnetzlistenformaten können im Bedarfsfall auf Anfrage implementiert bzw. bereitgestellt werden.

Bartels AutoEngineer FabView

Als preisgünstiger Viewer mit Ausgabemöglichkeit von Produktionsdaten wird **Bartels AutoEngineer FabView** angeboten. Diese Version ist für Fertigungsabteilungen vorgesehen, die diverse Datenausgaben und Druckausgaben erzeugen müssen, das Layout selbst aber nicht editieren. **Bartels AutoEngineer FabView** kann sowohl mit **BAE Professional** als auch mit **BAE HighEnd** betrieben werden und bietet die gleiche Funktionalität, lediglich das Speichern von Layoutdesignänderungen in BAE-Projektdateien ist deaktiviert.

1.1.2 Systemkomponenten

Der **Bartels AutoEngineer** besteht im Wesentlichen aus einem Paket zur Schaltplanerstellung, einem grafischen **Layouteditor** mit **Autoplacement** und **Autorouter**, sowie einem **CAM-Prozessor** mit einem Zusatzmodul zur Visualisierung und Bearbeitung von CAM-Daten. Der Designfluss zwischen Stromlauf und Layout (Forward- bzw. Backward-Annotation) wird durch ein eigenständiges **Packager**-Modul und eine Backannotation-Funktion im **Schaltplaneditor** realisiert. BAE unterstützt den Entwickler somit von der Generierung des Schaltplans über automatisches Platzieren und Routen bis hin zur Erstellung aller notwendigen Fertigungsunterlagen einschließlich der dazugehörigen Steuerdaten. Abbildung 1-1 zeigt das Designflussdiagramm des **Bartels AutoEngineer**.

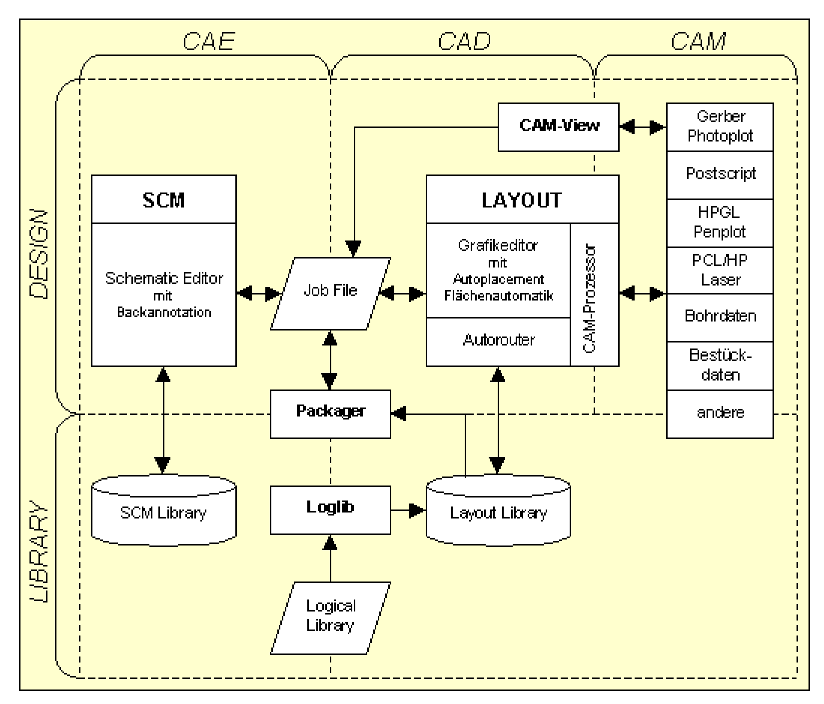


Abbildung 1-1: Bartels AutoEngineer Designfluss-Diagramm

1.1.3 Datenbankstruktur

Da in der CAD-Technik mit großen Datenmengen gearbeitet werden muss, ist die Mächtigkeit und Funktionalität der verwendeten Datenbank mit das wichtigste Kriterium zur Beurteilung eines CAx-Systems. Der **AutoEngineer** basiert auf einer speziell entworfenen, homogenen, objektorientierten, hierarchischen Datenbankstruktur mit einem optimierten B-Tree Suchbaum-Algorithmus. Durch die variable Schlüsselwortlänge und die Verwendung generischer Datenstrukturen weist der **AutoEngineer** bei größtmöglicher Redundanzfreiheit keine softwarebedingten Systemgrenzen auf. Die Systemgrenzen ergeben sich lediglich aus dem zur Verfügung gestellten Speicherplatz. Aus diesem Grunde gibt es keine Begrenzung in der Anzahl der Stromlaufblätter je Stromlaufplan, Anzahl der Schaltzeichen je Stromlauf, Anzahl der Bauelemente je Layout, Anzahl der Lötaugen je Bauteil oder Anzahl der Verbindungen usw.

Die Verwaltung der CAD-Daten erfolgt über eine integrierte Datenbank mit einem für das gesamte System einheitlichen Binärdateiformat. Dabei wird ein CAD-Objekt, zum Beispiel eine Leiterkarte, dynamisch aus ihren einzelnen Elementen zusammengesetzt, wobei diese ebenso aufgebaut werden. Um nun zu vermeiden, dass Änderungen in den Bibliotheken ungewollt auf bereits bestehende Designs einwirken, werden zum Zeitpunkt des ersten Zugriffs, zum Beispiel bei der Platzierung, alle benötigten Elemente einschließlich ihrer Unterelemente aus der Bibliothek in die Datei kopiert. Dadurch ist es unter anderem möglich, ein Objekt aus mehreren Bibliotheken oder eine Bibliothek mit Hilfe einer anderen Bibliothek aufzubauen. Das Konzept impliziert ebenfalls, dass Bibliotheken mit den gleichen Funktionen wie normale CAD-Dateien bearbeitet werden können. Faktisch kann sogar eine Projektdatei als Bibliothek für eine andere Projektdatei fungieren. Auch nachträgliche globale Änderungen z.B. eines Pintyps (im SCM) oder eines Bauteiltyps (im Layout) stellen kein Problem dar.

Die Projektdateien des **Bartels AutoEngineer** enthalten alle jeweils für ein Projekt relevanten Informationen. So können sämtliche Designdaten für ein beliebig großes Projekt (Stromlaufplan, Leiterplatten-Layout, Netzliste, Bibliothek, Steuerdaten) in einer einzigen Datei gespeichert werden, die damit das komplette Projekt beschreibt. Durch den dadurch gleichzeitig sichergestellten konsistenten Aufbau jobspezifischer Bibliotheken innerhalb der Projektdateien ergeben sich immense Vorteile z.B. hinsichtlich der Archivierung von Projekten oder des Datenaustauschs zwischen verschiedenen Firmen bzw. Abteilungen.

1.1.4 Datentypen und Bedienungskonzept

Das System ist praxisnah und einfach zu bedienen. Die Menüs und Handbücher sind wahlweise in verschiedenen Sprachen (Deutsch, Englisch, etc.) erhältlich. Die Standard-Benutzeroberfläche des **Bartels AutoEngineer** ist auf allen Hardware- bzw. Betriebssystemplattformen identisch und gewährleistet durch weit reichende Analogien bei der Generierung oder Bearbeitung von Objekten jeglicher Art eine leichte Erlernbarkeit. Die Windows-Versionen der BAE-Software bieten die Möglichkeit, wahlweise die BAE-Windows-Benutzeroberfläche mit Pulldownmenüs zu verwenden. Darüber hinaus kann die Benutzeroberfläche des **Bartels AutoEngineer** mit Hilfe mächtiger Tools zur Online-Tastaturprogrammierung bzw. zur Menübelegung praktisch beliebig an anwenderspezifische Bedürfnisse angepasst werden.

Durch die Verwendung von Fließpunktarithmetik (relative Genauigkeit besser als 1/10.000.000) entfallen praktisch alle Rasterbeschränkungen. Eine Undo/Redo-Funktion gewährleistet Datensicherheit und ermöglicht die komfortable Überprüfung von Realisierungsalternativen. Extrem hohe Geschwindigkeit bei der Ausführung grafischer Interaktionen (Bildaufbau bei Zoom/Pan, Bewegen von Grafikelementen, usw.) ist ebenso selbstverständlich wie die Bereitstellung grafischer Hilfsmittel (definierbare Farbtabellen, Anzeige, Abfrage, Highlight von Elementen, Verbindungen, Fehlern, usw.).

Im Speicher werden die Daten über einen Pool verwaltet, der auch Funktionen wie Undo und die Bildschirmausgabe steuert. Da die Daten in Echtzeit aus der hierarchischen Darstellung in die Vektor- und Polygondarstellung zum Bildaufbau konvertiert werden, lassen sich auch komplexe Operationen wie die Verschiebung von Bauteilen schnell durchführen. Um eine hohe Bildaufbaugeschwindigkeit zu gewährleisten, wird z.B. vor der Ausgabe komplexer Bauteile grundsätzlich überprüft, ob das Bauteil überhaupt im Bildschirmbereich liegt. Diese Abfragen erfolgen aber über ein wohldefiniertes Interface, um die Anpassbarkeit an die unterschiedlichsten Grafikkontroller zu gewährleisten. Umfangreiche Objekte wie Leiterbahnzüge und Flächen werden natürlich in komprimierter Form abgespeichert. So werden bei einem Leiterzug nur der erste, die Eckpunkte und der letzte Punkt abgespeichert, und nicht jedes Segment mit Anfangs- und Endpunkt einzeln, um eine speicherverschlingende Redundanz zu vermeiden. Der gesamte auf einer Lage befindliche Leiterzug ist ein Poolelement. Auch ein komplettes PGA-Bauteil mit 84 Pins belegt bei Folgeplatzierungen nur ein Poolelement. Lediglich die Verdrahtungsinformation wird noch getrennt gespeichert, die Geometrie wird von der ersten Platzierung übernommen. Die maximale Anzahl Poolelemente liegt bei 32-Bit Systemen bei 2^31=2.147.483.648. Somit dürfte der Pool bei nichtvirtuellen Systemen eher durch den zur Verfügung stehenden Speicher oder durch den Zeigeradressraum begrenzt sein. Aufgrund der geringen Redundanz sind jedoch selbst bei PC-Systemen mit eingeschränktem Hauptspeicher kaum Probleme zu erwarten.

Koordinaten und Winkel werden grundsätzlich intern im Fließpunktformat abgelegt. Dadurch kann jedes Element auf beliebigen Koordinaten exakt platziert und zusätzlich um einen beliebigen Winkel gedreht werden. Immerhin überschreitet die Genauigkeit von 32-Bit IEEE Fließpunktzahlen die von 16-Bit Integerzahlen, die in "Standard"-Systemen im Allgemeinen verwendet werden. Durch die geschickte Einbindung in das hierarchische System entstehen selbst bei der Verwendung von langsamen Standard-Numerikprozessoren kaum Rechenzeitnachteile. Operationen, die eine erhöhte Genauigkeit erfordern, werden grundsätzlich mit doppelter (64-Bit) Präzision ausgeführt. Referenzen auf andere Elemente (wie z.B. Pinnamen) werden grundsätzlich als Zeichenketten abgespeichert, wobei identische Zeichenketten pro Dateieintrag zu einer zusammengefasst sind. Somit ist die richtige Bezeichnung von Stecker- und PGA-Pins mit Namen wie **C32** problemlos möglich.

1.1.5 Schnittstellen zu anderen Systemen

Bestehende Applikationen sind leicht an- bzw. einbindbar. So existieren z.B. Werkzeuge zur Übernahme unterschiedlicher ASCII-Netzlistenformate. Definierte Schnittstellen zur Fertigung (Bestückdaten, Bohrdaten, Gerber-Fotoplot, HP-GL-Penplot, Postscript, usw.) werden über den **CAM-Prozessor** angeboten und können innerhalb der Benutzeroberfläche komfortabel in auf Knopfdruck startbare Batchabläufe für die Komplettausgabe aller Fertigungsdaten integriert werden. Zur Dokumentation von Schaltplan und Layout stehen frei konfigurierbare PDF-Ausgaben zur Verfügung. Eine speziell entwickelte Programmiersprache (**User Language**) bietet dem Anwender wahlfreien Zugriff auf alle Designdaten; damit lassen sich kundenspezifische Programme zur Ausgabe von Stücklisten, Netzlisten, Geometriedaten, Bohrdaten, Bestückdaten, Fräsdaten, usw. in frei definierbaren Formaten implementieren.

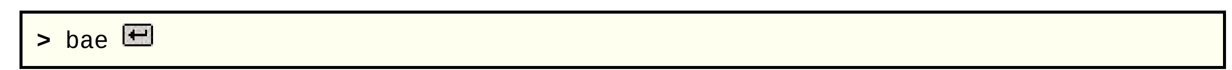
1.2 Grundsätzliches zur Bedienung

1.2.1 Programmaufruf und Bildschirmaufbau

BAE-Aufruf

Zum Arbeiten mit dem **Bartels AutoEngineer** legen Sie, sofern nicht schon bei der Installation geschehen, ein Verzeichnis für Ihre Daten bzw. Projektdateien an.

Sie können den **Bartels AutoEngineer** dann in diesem Verzeichnis durch Eingabe des Befehls



```
> bae
```

starten.

Unter Windows kann der **Bartels AutoEngineer** auch durch Selektion der Datei **`bae.exe`** über die Funktion Ausführen im Dateimenü des Programm-Managers gestartet werden. Darüber hinaus bieten Windows- bzw. X11/Motif-basierende Betriebssysteme die Möglichkeit, Applikationen wie den **Bartels AutoEngineer** durch Anklicken des Piktogramms der entsprechenden Programmdatei zu starten. Dabei kann auch das Verzeichnis festgelegt werden, in dem die Applikation gestartet werden soll, und schließlich lässt sich der BAE-Aufruf durch die Einbindung einer Referenz des BAE-Startup-Icons in das sogenannte Launchpad weiter vereinfachen. Außerdem wird die **`.ddb`**-Dateiextension so mit dem BAE verknüpft, dass beim Doppelklick auf eine **`.ddb`**-Datei automatisch der **Layouteditor** gestartet und das Defaultlayout in den Speicher geladen wird. Mit dem bei Rechtsklick auf eine **`.ddb`**-Datei erscheinenden Kontextmenü kann wahlweise der **Layouteditor** oder der **Schaltplaneditor** (mit Laden des neuesten Schaltplans) gestartet werden. Nähere Informationen zur Konfiguration von Applikationsaufrufen entnehmen Sie bitte der Dokumentation Ihres Betriebssystems.

BAE-Benutzeroberfläche mit Pulldownmenü (Standard)

Die Windows- und Motif-Versionen des **Bartels AutoEngineer** werden per Default mit einer Benutzeroberfläche mit Pulldownmenüs installiert. Diese Benutzeroberfläche ist vertikal unterteilt und besteht aus der Hauptmenüleiste oben, dem Grafikarbeitsbereich in der Mitte, sowie einer Infozeile und der Eingabe- und Mitteilungszeile unten. Bei Selektion einer Hauptmenüfunktion wird das entsprechende Funktionsmenü im Pulldownmodus aktiviert. In der Windows-Version kann mit Hilfe der Funktion Menubaum aus dem Untermenü Werkzeugleiste des Menüs Ansicht wahlweise links oder rechts vom Grafikarbeitsbereich ein Menübaum mit explorerartiger Darstellung der Menüstruktur eingeblendet werden.

Im Grafikarbeitsbereich steht direkt nach dem Aufruf von BAE das Bartels-Firmenlogo bzw. ein Copyright-Vermerk, in der Mitteilungszeile die Angabe der Programmversion bzw. des Benutzernamens.

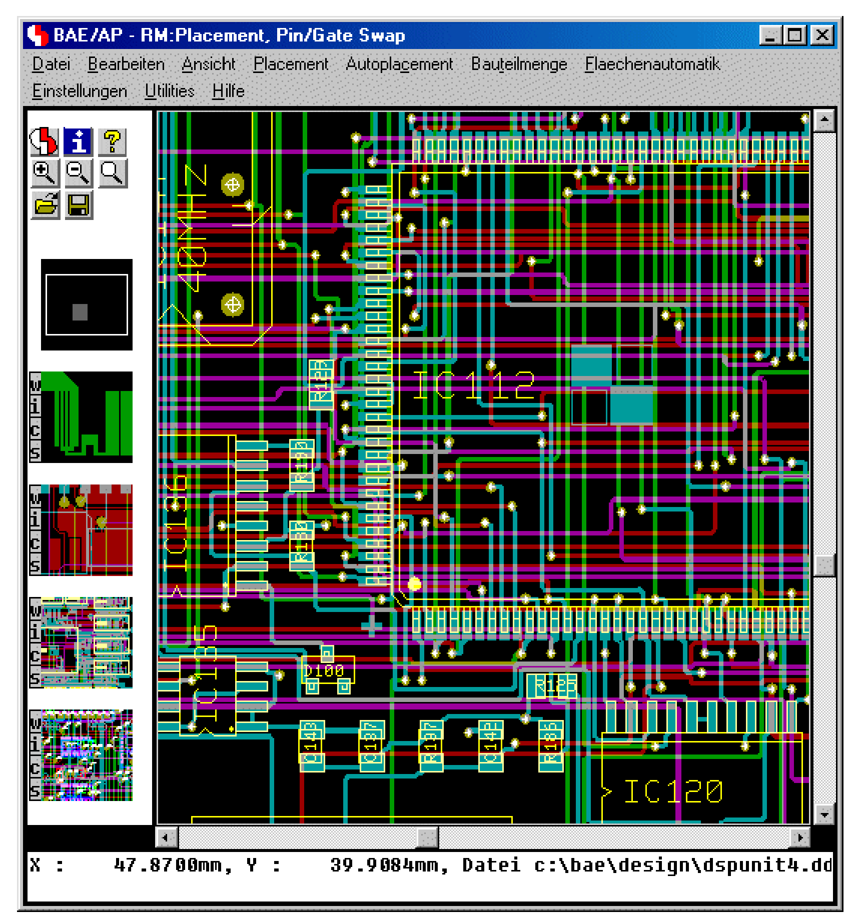


Abbildung 1-2: BAE-Benutzeroberfläche mit Pulldownmenü

Sollte nach dem Aufruf des BAE die Meldung **`Die Benutzungsberechtigung fehlt!`** erscheinen, dann deutet dies auf eine fehlerhafte Installation der Software hin. Überprüfen Sie in diesem Fall, ob die Software richtig autorisiert ist (durch Hardlock Key und Installation der entsprechenden Lizenzdatei; siehe hierzu die Bartels AutoEngineer® Installationsanleitung).

BAE-Seitenmenü-Benutzeroberfläche

Alternativ kann der BAE mit einer Seitenmenü-Benutzeroberfläche betrieben werden. Diese besteht aus einem Info-Feld und Menüfeldern an der rechten Seite, links davon dem eigentlichen Grafikarbeitsbereich sowie unterhalb des Grafikarbeitsbereichs einer Eingabe- und Mitteilungszeile.

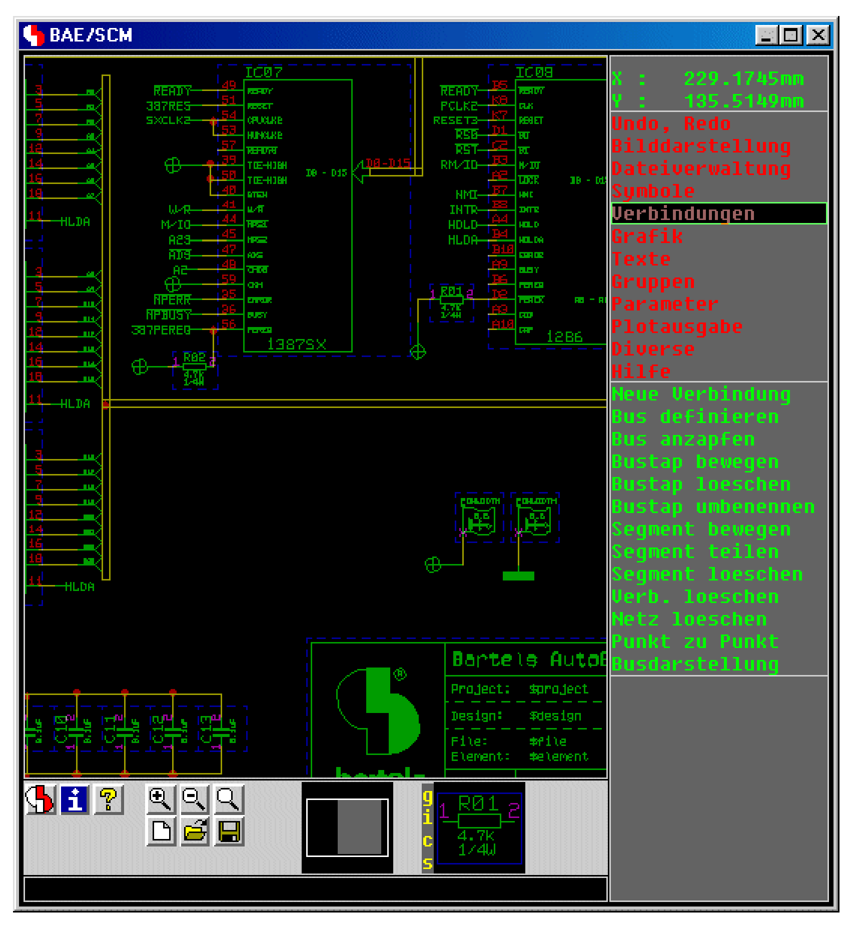


Abbildung 1-3: BAE-Benutzeroberfläche mit Seitenmenü

Die Aktivierung der BAE-Benutzeroberfläche mit Seitenmenüs in den Windows- bzw. Motif-Versionen der BAE-Software erfolgt über das Setup aus dem BAE-Hauptmenü oder mit Hilfe des Utilityprogramms **BSETUP** (siehe hierzu auch Kapitel 7.2).

Sollte nach dem Aufruf der BAE-DOS-Version der Grafikaufbau fehlschlagen, dann deutet dies auf eine fehlerhafte Installation der Software hin. Überprüfen Sie in diesem Fall, ob der (richtige) Grafiktreiber installiert ist (siehe hierzu die Bartels AutoEngineer® Installationsanleitung).

1.2.2 Funktionsauswahl

BAE-Windows/Motif-Benutzeroberfläche

Am oberen Fensterrand der BAE-Benutzeroberflächen mit Pulldownmenüs ist ständig eine Hauptmenüleiste verfügbar. Durch die Anwahl eines Hauptmenüpunktes wird in der Regel ein Pulldownmenü mit spezifischeren Funktionen aktiviert. Eine Menüfunktion wird durch Verschieben der Maus auf den entsprechenden Menüeintrag und Betätigen der linken Maustaste aktiviert. Üblicherweise enthalten die Menüeinträge unterstrichene Zeichen, durch die sogenannte Hotkeys zur schnellen Aktivierung bzw. Selektion des entsprechenden Menüs bzw. der entsprechenden Funktion gekennzeichnet sind. D.h., Hauptmenüfunktionen können wahlweise durch Drücken des gekennzeichneten Zeichens zusammen mit der Alt-Taste aktiviert bzw. selektiert werden, und Pulldownmenüfunktionen lassen sich einfach durch Betätigen der gekennzeichneten Taste aktivieren. Die mittlere Maustaste hat eine Sonderfunktion; mit ihr können Sie jederzeit, d.h. auch aus beliebigen Funktionen heraus ein besonderes Untermenü mit häufig benötigten Funktionen zur Änderung der Bilddarstellung anwählen. Durch Betätigung der mittleren Maustaste wird in der Mitteilungszeile der aktuelle Datei- und Elementname angezeigt, sofern nicht gerade eine andere Menüfunktion aktiv ist. Die Betätigung der mittleren Maustaste lässt sich auch durch simultanes Drücken der linken und der rechten Maustaste simulieren. Somit kann das Online-Bilddarstellungsmenü auch jederzeit aktiviert werden, wenn nur eine Zweitasten-Maus installiert bzw. konfiguriert ist. Durch Betätigen der ESC-Taste (ASCII-Code 27) kann die aktuell aktive Menüfunktion jederzeit abgebrochen werden (Abort Hotkey).

Wenn Sie zum Beispiel im **Layouteditor** die Funktion Neues Bauteil erreichen wollen, so wählen Sie zunächst den Menüpunkt Bauteile im Hauptmenü und danach die im Menü Bauteile erscheinende Funktion Neues Bauteil an. Unmittelbar nach Beendigung einer Funktion können Sie diese über die linke Maustaste erneut aktivieren. Die aktuell der linken Maustaste zugewiesene Funktion wird in der Titelleiste des BAE-Fensters angezeigt. Über Betätigung der H-Schaltfläche der Toolbar mit der rechten Maustaste stehen die letzten 16 aufgerufenen Menüpunkte im Schnellzugriff zum erneuten Aufruf zur Verfügung.

Die Menüs in den Windows- und Motif-Versionen sind kontextsensitiv. D.h., die einzelnen Menüfunktionen sind nur bei tatsächlicher Eingabebereitschaft bzw. Anwendbarkeit selektierbar ("Ghostmenüs"). Zur besseren Übersichtlichkeit sind die Menüs zudem durch Verwendung von Trennlinien in logische Funktionsgruppen unterteilt.

In den Windows- und Motif-Versionen kann durch Anklicken eines Elementes im Arbeitsbereich mit der rechten Maustaste ein Kontextmenü mit für dieses Element sinnvollen Funktionen aktiviert werden. Die p-Taste (Properties) aktiviert eine Dialogbox in der die Eigenschaften des unter dem Mauszeiger befindlichen Elementes angezeigt werden und auch geändert werden können.

Spezielle Funktionen wie z.B. die Dateinamensauswahl in den Windows- bzw. Motif-Benutzeroberflächen der BAE-Software sind mit Windows- bzw. Motif-spezifischen Popupmenüs bzw. Dialogen anstelle der BAE-Standard-Popupmenüs ausgestattet. Unter Windows und Motif wird bei längeren Textausgaben im Arbeitsbereich der Zugriff auf den gesamten ausgegebenen Text über Scrollbars unterstützt.

Unter Windows und Motif kann der aktuell sichtbare Bildausschnitt mit Hilfe der Kursor- bzw. Pfeiltasten um jeweils die halbe Anzeige in die tastenspezifische Richtung bewegt werden. Die Tasten PageUp und PageDown scrollen um die volle Bildhöhe nach oben bzw. unten, bei gleichzeitig gedrückt gehaltener Umschalt/Shift-Taste entsprechend nach rechts bzw. links. Der Bildauschnitt lässt sich maximal bis zu den Grenzen des aktuell geladenen Elements bewegen. Mit den Tasten Home/Pos1 und End/Ende kann direkt zur oberen bzw. unteren Elementgrenze gesprungen werden, bei gleichzeitig gedrückt gehaltener Umschalt/Shift-Taste entsprechend zur linken bzw. rechten Elementgrenze.

Unter Window und Motif kann der aktuell sichtbare Bildausschnitt auch mit Hilfe des Mausrades verändert werden. Beim Drehen des Mausrades wandert der Bildausschnitt jeweils um die halbe Anzeigebreite nach oben- bzw. unten, bei gleichzeitig gedrückt gehaltener Umschalt/Shift-Taste entsprechend nach rechts bzw. links. Wird bei gedrückt gehaltener linker Maustaste am Mausrad gedreht, so wird der aktuelle Bildschirmausschnitt je nach Drehrichtung vergrößert oder verkleinert.

In den Windows- und Motifversionen ist ein Programmende über das Systemmenü des BAE-Fensters, bzw. durch Anklicken des Windows-Buttons zum Schließen der Applikation möglich. Beachten Sie jedoch, dass diese Beendigungsprozeduren ggf. eine Bestätigung erfordern, um einem versehentlichen Verwurf von Designänderungen vorzubeugen. Wahlweise kann dabei das aktuell bearbeitete Element gesichert werden.

In der Windowsversion wird bei Mausselektion eines Menüpunktes mit gleichzeitig gedrückt gehaltener Umschalt/Shift-Taste die Hilfe zu zu diesem Menüpunkt aktiviert, für die Wiederholfunktion der linken Maustaste aber der selektierte Menüpunkt gespeichert. So kann nach dem Studium der Hilfe die entsprechende Funktion durch einfache Betätigung der linken Maustaste im Arbeitsbereich aufgerufen werden, ohne dass erneut im Menü zum Menüpunkt navigiert werden muss.

Beim Beenden des BAE unter Windows und Motif werden die aktuelle Position und Größe (und ggf. der Vollbildmodus) der BAE-Applikationsfenster sowie die Dimensionen und Positionen funktionsspezifischer Dialogboxen modulspezifisch in einer Konfigurationsdatei mit dem Namen `baewin.dat` bzw. `baexwin.dat` im BAE-Programmverzeichnis gespeichert. Beim nächsten BAE-Aufruf wird diese Konfiguration automatisch wiederhergestellt. In **BAE HighEnd** sind die gespeicherten Positionseinträge neben dem BAE-Modulnamen auch noch mit der Nummer des Fensters der aktuellen Sitzung verknüpft, so dass z.B. die Position von zwei parallel geöffneten **Schaltplaneditor**-Fenstern einer Sitzung bei der nächsten Sitzung wiederhergestellt werden kann. In diesem Zusammenhang ist ein Mehrschirmbetrieb des **BAE HighEnd** besonders zu empfehlen.

BAE-Seitenmenü-Benutzeroberfläche

Das rechte Menü der BAE-Seitenmenü-Benutzeroberfläche ist immer in zwei Hälften unterteilt, wobei Sie zum Start einer Funktion immer beide Hälften anwählen können. In der oberen Hälfte befindet sich das Hauptmenü mit den Menüpunkten zur Auswahl des in der unteren Hälfte angezeigten Menüs. Dadurch erreichen Sie alle Funktionen des selben Arbeitsgangs mit nur einer und alle anderen mit maximal zwei Maustastenbetätigungen.

In den Menüfeldern der BAE-Standard-Benutzeroberfläche befindet sich ein grüner Balken, der durch Bewegen der Maus verschoben werden kann. Durch Betätigen der linken oder rechten Maustaste wird der entsprechende Menüpunkt angewählt. Wenn Sie zum Beispiel im **Layouteditor** die Funktion Neues Bauteil erreichen wollen, so wählen Sie zunächst den Menüpunkt Bauteile im Hauptmenü und danach die nunmehr im Menü Bauteile erscheinende Funktion Neues Bauteil an. Nach Beendigung dieser Funktion können Sie diese durch Drücken der rechten oder linken Maustaste erneut aktivieren oder z.B. die im gleichen Menü befindliche Funktion Loeschen Bauteil direkt anwählen. Ein Wechsel zu der Funktion Neue Leiterbahn bedarf aus diesem Menü heraus nur der Anwahl der Punkte Leiterbahnen im Hauptmenü und dann direkt der Funktion Neue Leiterbahn im Menü Leiterbahnen.

Nach Anwahl einer Funktion erscheint entweder ein weiteres Menü, die Aufforderung zu einer Eingabe in der Eingabe- und Mitteilungszeile, oder aber ein Fadenkreuz im eigentlichen Grafikarbeitsbereich. In der Statuszeile werden ggf. Hilfsinformationen oder Prompts zur angewählten Funktion eingeblendet. Funktionsspezifische Eingabeaufforderungen (z.B. für Koordinaten- und Längen- und Breitenangaben) und Fehlermeldungen enthalten soweit vorhanden den Namen des betroffenen bzw. bearbeiteten Elements. Bei Eingaben über den Grafikkursor erscheint in der Mitteilungszeile ein Hinweis zu der erwarteten Eingabe. Der Menübalken wird üblicherweise rot dargestellt, wenn das System auf Benutzereingaben wartet. Die mittlere Maustaste hat eine Sonderfunktion; mit ihr können Sie jederzeit, d.h. auch aus beliebigen Funktionen heraus ein besonderes Untermenü mit häufig benötigten Funktionen zur Änderung der Bilddarstellung anwählen. Durch Betätigung der mittleren Maustaste wird in der Mitteilungszeile der aktuelle Datei- und Elementname angezeigt, sofern nicht gerade eine andere Menüfunktion aktiv ist. Die Betätigung der mittleren Maustaste lässt sich auch durch simultanes Drücken der linken und der rechten Maustaste simulieren. Somit kann das Online-Bilddarstellungsmenü auch jederzeit aktiviert werden, wenn nur eine Zweitasten-Maus installiert bzw. konfiguriert ist. Durch Betätigen der **ESC**-Taste (ASCII-Code 27) kann die aktuell aktive Menüfunktion jederzeit abgebrochen werden (Abort Hotkey).

Eine ganze Reihe von Funktionen erlauben wahlweise die mausgesteuerte Selektion der auszuführenden Funktion bzw. des zu bearbeitenden Elements über Popupmenüs. Dadurch vereinfacht sich die Bedienung elementarer Datenverwaltungsfunktionen wie Laden Element, Loeschen Element, Dateiinhalt, Farben laden, usw. ganz erheblich. Grundsätzlich ist parallel zur Popupmenüauswahl jeweils auch immer die manuelle Eingabe des Elementnamens über die Mitteilungszeile möglich. Auch enthält jedes Popupmenü nach Bedarf spezielle Buttons wie Abbruch (zum Abbrechen der Funktion), Weiter (zum Weiterblättern in der Menüauswahl) oder Zurueck (zum Zurückblättern in der Menüauswahl).

In den Dialogen bzw. Popupmenüs zur Auswahl von Bauteil- und Netznamen aus der Netzliste kann durch Eingabe von `?prefix` zu einem Namensprefix in der angezeigten Liste gesprungen werden. So positioniert z.B. `?r4` den sichtbaren Listenausschnitt auf den ersten mit `r4` beginnenden Namen, bzw. falls kein solcher vorhanden ist, auf den nächsten nach `r4` folgenden Namen. Bei aufeinanderfolgenden Namensabfragen bleibt die aktuelle Position der Anzeige innerhalb der Auswahlliste erhalten, und es muss nicht erst wieder zu dem zuvor selektierten Listenauschnitt geblättert werden.

Die Farbdarstellung der BAE-Seitenmenü-Benutzeroberfläche kann mit Hilfe des Utilityprogramms **BSETUP** an die anwenderspezifischen Bedürfnisse angepasst werden. Durch eine geeignete Einstellung der Menüfarben lässt sich u.U. ein nicht zu unterschätzender ergonomischer Vorteil erzielen (Erkennung des aktiven Menüs bzw. der aktiven Funktion aus dem Augenwinkel). Eine genaue Beschreibung des Programms **BSETUP** finden Sie im Kapitel 7.2 dieses Handbuchs.

Anpassungen der BAE-Benutzeroberfläche

Über die **Bartels User Language** werden Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Damit ist es möglich, automatisierte **User Language**-Programmaufrufe auf die Tastatur zu legen (z.B. Taste r zur Aktivierung des **User Language**-Programms **ROTATE**). Durch die Definition von Toolbars bzw. die Zuweisung spezieller **User Language**-Programmaufrufe auf neue bzw. bestehende Menüs oder Menüeinträge können die Menüoberflächen der **AutoEngineer**-Module mit integriertem **User Language Interpreter** praktisch nach Belieben konfiguriert werden. Die Tastaturprogrammierung bzw. die Menübelegung lässt sich vollautomatisch über die jeweiligen **User Language**-Startupprogramme durchführen. Mit einem entsprechenden **User Language**-Programm ist sogar die dynamische Änderung der Tastatur- und Menübelegung während der Bearbeitung möglich. Damit besteht prinzipiell völlige Freiheit in der Konfiguration der Benutzeroberflächen der **AutoEngineer**-Module mit integriertem **User Language Interpreter** Beachten Sie daher bitte, dass Ihre aktuell definierte Benutzeroberfläche unter Umständen anwenderspezifische Zusatzfunktionen anbietet, die in dieser Dokumentation nicht erläutert sind. Eine ausführliche Beschreibung der **Bartels User Language** und die Möglichkeiten des impliziten **User Language**-Programmaufrufs finden Sie im Bartels User Language Programmierhandbuch.

Die Benutzeroberflächen der Windows- und Motifversionen des **Bartels AutoEngineer** ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Windows- und Motifmenüs der BAE-Module werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs konfiguriert. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist ebenfalls entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

In den Windows- und Motif-Versionen der BAE-Software sind eine Reihe von Dialogen für Parametereinstellungen implementiert. Hierzu zählen Dialoge zur Einstellung von Bilddarstellungsparametern und zur Konfiguration allgemeiner Bearbeitungsparameter für alle BAE-Module, ein Dialog zur Einstellung von SCM-Plotparametern, Dialoge für **Autoplacement**- und Flächenfüllparameter, Dialoge zur Einstellung von Routingoptionen, Routingstrategie, Routersteuerung sowie zur Festlegung von Router-Batchläufen im **Autorouter**, Dialoge zur Einstellung von Parametern für Kontrollplot, Gerberphotoplot und Bohrdatenausgabe im **CAM-Prozessor**. Diese Dialoge können über die **User Language**-Systemfunktion **bae_callmenu** aktiviert werden. In den Windows- und Motif-Modulen der BAE-Software konfiguriert das **User Language**-Programm **UIFSETUP** Menüfunktionen zum Aufruf der Dialoge.

BAE HighEnd Message-System

In die **BAE HighEnd**-Version des **Bartels AutoEngineer** ist ein Message-System integriert, über das die einzelnen Programm-Module des **AutoEngineer** miteinander kommunizieren können. Die Vermittlerfunktion übernimmt dabei das zum Programmstart aufgerufene BAE-Modul. Damit das dadurch aktivierte Message-System genutzt werden kann, sind weitere Modul-Aufrufe nur über die hierfür in **BAE HighEnd** zur Verfügung stehenden Menüfunktionen vorzunehmen (Weitere Task im BAE-Hauptmenü, Neues SCM Fenster in den Utilities-Menüs des **Schaltplaneditor** und **Layouteditor** bzw. Wechsel zwischen verschiedenen Programm-Modulen der aktuellen AutoEngineer-Sitzung). Durch den Einsatz von Multitasking und Pipes ist in **BAE HighEnd** die simultane Darstellung unterschiedlicher Ansichten eines Designs (Gesamtübersicht, Ausschnittvergrößerung) ebenso möglich wie das gleichzeitige Arbeiten an Schaltplan und Layout. Damit werden in **BAE HighEnd** automatisch auch modulübergreifende Features wie z.B. simultanes bzw. globales Netzhighlight für Schaltpläne und Layout eines Designs (projektspezifisches Multi-Windowing/Multi-Tasking, Cross-Probing) unterstützt.

1.2.3 Elementare Funktionen

Ansicht, Bilddarstellung

Mit der mittleren Maustaste können Sie jederzeit das Menü Ansicht auch inmitten einer grafischen Eingabe erreichen. So lässt sich z.B. der Pick eines Bauteils und die Grobplatzierung in der Layoutübersicht, hingegen die Feinplatzierung und Ablage nach Auswahl eines detaillierteren Bildausschnitts durchführen. Nach der Beendigung einer Funktion aus dem so selektierten Menü Ansicht kommen Sie stets wieder an die richtige Stelle der vorher gewählten Funktion zurück. Im Menü Ansicht finden Sie auch andere wichtige Funktionen wie z.B. die Farbauswahl und die Wahl des Eingaberasters.

Farbauswahl

Die Farbauswahl erfolgt zunächst über die Funktion Farbpalette im Menü Ansicht, jedoch können verschiedene Farbeinstellungen komplett unter einem Namen auf der Festplatte abgespeichert werden (Farben speichern) und dann mit Farben laden wieder aufgerufen werden. Auch diese Funktion ist über die mittlere Maustaste jederzeit erreichbar. Bei Überlappungen verschiedener Elemente werden grundsätzlich die resultierenden Mischfarben dargestellt. Die mit Highlight gewählte Farbe wird ebenfalls mit der Farbe des zu markierenden Elements gemischt und ergibt dann die neue hellere Elementfarbe.

In den Farbauswahlmenüs erfolgt die Zuweisung einer Farbe an einen speziellen Anzeigeelementtyp durch Selektion des Anzeigeelements (bzw. der Lage) über die linke Maustaste sowie die anschließende Selektion der gewünschten Farbe. In den Farbauswahlmenüs des Layoutsystems besteht zusätzlich die Möglichkeit der schnellen Lagen-Ein/Ausblendung mit Erhalt der aktuell eingestellten Farbe. Die Aktivierung bzw. Deaktivierung der Lagenanzeige erfolgt dabei durch Anwahl des Farbbuttons der gewünschten Lage mit der rechten Maustaste. In der Menüanzeige werden die Farbbuttons der aktuell ausgeblendeten Lagen durchgestrichen dargestellt.

Eingaberaster

Die Wahl des richtigen Eingaberasters ist von grundlegender Bedeutung z.B. für die spätere einfache Entflechtung und Fertigung des Layouts. Mit der Platzierung von Bauteilen in einem 1/10" oder 1/20" Raster mit nur ausnahmsweiser Abweichung (z.B. bei Steckern) erleichtern Sie sich sowohl die Handverlegung von Leiterbahnen als auch dem **Autorouter** die automatische Entflechtung. Auch der Bestücker Ihrer Platinen hat es dadurch wesentlich leichter. Grundsätzlich ist jedoch das Bartels Layoutsystem durch die Verwendung einer Fließpunkt-Datenbasis weder an bestimmte Raster, noch an bestimmte Winkel gebunden (vgl. Option Winkel/Raster freigeben). Trotzdem empfehlen wir z.B. auch die Verlegung von Leiterbahnen in 45-Grad Winkelschritten, sofern zur Abweichung hiervon keine Notwendigkeit besteht. Die Einhaltung bzw. Freigabe von Fangwinkel und Fangraster sowie das aktuelle Eingaberaster können jederzeit über das mit der mittleren Maustaste erreichbare Bilddarstellungsmenü verändert werden.

Der Mauszeiger springt bei gedrückt gehaltener **Umschalt/Shift**-Taste mit Hilfe der Kursor- bzw. Pfeiltasten jeweils zum nächsten Rasterpunkt in der tastenspezifischen Richtung. Mit **Umschalt/Shift** und **Eingabe/Enter** wird ein Rasterpunkt analog zum Mausklick mit der linken Maustaste selektiert und ein einfaches **Eingabe/Enter** beendet die Eingabe einer Punktliste analog zu rechter Maustaste und Fertig. Somit können im Raster verlaufende Leiterbahnen und Polygone auch rein über Tastatur gezeichnet werden.

Koordinateneingaben

Bei Eingaben im Grafikarbeitsbereich besteht in den meisten mit der rechten Maustaste erreichbaren Untermenüs die Möglichkeit der direkten Koordinateneingabe. Diese wird im Allgemeinen mit den Optionen Sprung relativ und Sprung absolut durchgeführt. Dabei bezieht sich "relativ" auf die letzte vorherige grafische Eingabe der Funktion (z.B. vorheriger Polygon- oder Leiterbahneckpunkt) und "absolut" auf den Plannullpunkt. Sofern der mit "relativ" einzugebende Punkt der erste einer Funktion ist, beziehen sich die Koordinaten auch auf den Plannullpunkt. Jede Koordinate kann sowohl metrisch als auch zöllig (Inch) eingegeben werden. Zur metrischen Eingabe reicht die einfache Eingabe der Zahl aus, Nachkommastellen werden nach einem Dezimalpunkt eingegeben. Bei der zölligen Eingabe ist an die Zahl das Doppelhochkomma-Zeichen **"** anzufügen. Auch hier können Nachkommastellen nach einem Dezimalpunkt eingegeben werden. Mit Hilfe des **BSETUP**-Kommandos **`USERUNITS`** (siehe hierzu auch Kapitel 7.2) kann das System veranlasst werden, Koordinateneingaben in Inch zu interpretieren; mm-Angaben sind hierbei dann durch Anfügen von **`mm`** an den Koordinatenwert möglich. Metrische und zöllige Koordinaten können beliebig gemischt werden; aufgrund der Fließkomma-Datenbasis bleibt die Genauigkeit im erforderlichen Rahmen auch tatsächlich erhalten. Des Weiteren besteht die Möglichkeit, Polarkoordinaten einzugeben. Hierzu ist zunächst der **`Polarkoordinaten`**-Button zu betätigen. Daraufhin erfolgt die Abfrage von Radius und Winkel durch das System. Bei der Spezifikation von Polarkoordinaten kann der Winkel ebenfalls mit Nachkommastellen und durch Nachstellung des Buchstabens **`R`** auch in Radiant angegeben werden.

Alle numerischen Eingabefelder in den Dialogboxen des BAE können einfache arithmetische Ausdrücke mit den vier Grundrechenarten und durch runde Klammern vorgegebene Rangfolge der Operationen auswerten. Wird am Ende eine

Eingabefeldes ein Gleichheitszeichen eingegeben, erfolgt unmittelbar die Berechnung und Übertragung des Ergebniswertes in das Eingabefeld. Ansonsten wird der Term beim Beendigen der Dialogbox ausgewertet. So lässt sich also z.B. ein Sprung relativ auch durch ein Sprung absolut durchgeführen, bei dem die relativen Sprungkoordinaten mit (**+**) an die in der Dialogbox angezeigten alten Koordinatenwerte angehängt werden.

Dateiverwaltung

Um die meisten Funktionen starten zu können, muss zunächst ein Element (Schaltplanblatt oder Leiterkartenlayout, Bibliothekssymbol, usw.) geladen sein. Ein Element wird durch Angabe der Hierarchieebene, des Dateinamens und des Elementnamens selektiert. Der Elementname bezeichnet dabei das Element innerhalb der Projektdatenbank. Der Ladevorgang wird durch die Eingabe des gewünschten Elementnamens aktiviert. Die Auswahl der Datei- und Elementnamen kann wahlweise über Popupmenü und Mausklick oder durch direkte manuelle Eingabe über Tastatur erfolgen. Wird bei der Eingabe über Tastatur ein leerer Dateiname angegeben (Betätigen der Eingabetaste ↵), so verwendet das System den Dateinamen des zuletzt im Speicher befindlichen Elements. Bei DDB-Dateinamensabfragen im SCM und im Layout kann zudem immer durch Eingabe von **!** die aktuell über das BAE-Setup (siehe Kapitel 7.2) definierte SCM- bzw. Layoutstandardbibliothek referenziert werden.

Bitte vergessen Sie nicht, vor dem Laden eines anderen Elements, dem Erzeugen eines neuen Elements oder dem Verlassen des Programms das aktuell im Speicher befindliche Element zu speichern. Zur Sicherheit wird vor dem Schließen eines ungesicherten Elements ein Popupmenü mit Optionen zum Sichern des modifizierten Elements oder zum Verwurf der Designänderungen aktiviert.

Beim Laden eines Elementes wird die Updatezeit des Elements mitgeladen. Beim Speichern des Elementes wird überprüft, ob das Element in der Datei nicht zwischenzeitlich geändert wurde (z.B. im Netzwerk von einem anderen Anwender). Ggf. erfolgt eine Bestätigungsabfrage, ob das Element trotz zwischenzeitlicher Änderung überschrieben werden soll.

In der BAE-Seitenmenü-Benutzeroberfläche kann über die Funktionen zur Dateinamensauswahl wahlweise ein intelligentes Popupmenü zur optionalen Anwahl von Verzeichnissen aktiviert werden. Hierzu ist der Button Dir. im aktuellen Dateinamensauswahlmenü anzuwählen. Mit dem **BSETUP**-Kommando **PROJROOTDIR** (siehe auch Kapitel 7.2) kann die Wurzel des bei der Directoryauswahl anzuzeigenden Verzeichnisbaumes voreingestellt werden. Default ist hierbei der Relativpfadname **.** des aktuellen Verzeichnisses. Die Popuphintergrundfarbe für die Verzeichnisauswahl ist mit der **FRAMECOLOR**-Option **POPMFILL** über **BSETUP** (siehe Kapitel 7.2) einstellbar. Verzeichnisse mit Unterverzeichnissen werden in einer Strukturanzeige mit hierarchisch angeordneten Grafikrahmen dargestellt. Verzeichnisse ohne Unterverzeichnisse werden am Ende der Auswahlliste durch den entsprechenden Verzeichnisnamen angezeigt. Verzeichnisse, in denen Dateien mit der Namenserweiterung **.ddb** (DDB-Dateien) enthalten sind, werden durch Anfügen eines Pluszeichens (**+**) an das Ende des Verzeichnisnamens gekennzeichnet. In den Verzeichnisauswahlmenüs kann im Bedarfsfall mit dem Button Weiter im Verzeichnismenü weitergeblättert werden. Mit dem Button ...Zoom kann in eine Detailanzeige des gewählten Verzeichnisses geschaltet werden; das Zurückschalten in die jeweilige Übersichtsdarstellung des aktuell selektierten Verzeichnisses erfolgt durch Anwahl des Buttons Parent. Mit dem Button Abbruch kann die übergeordnete Dateinamensabfrage aus dem Verzeichnisauswahlmenü heraus abgebrochen werden. Die Fehlermeldung **Keine Verzeichnisse hier gefunden!** besagt, dass in dem über **PROJROOTDIR** eingestellten Verzeichnis keine Unterverzeichnisse enthalten sind. Nach Selektion eines gültigen Verzeichnisnamens wird ein Menü mit den Dateinamen des gewählten Verzeichnisses angezeigt.

In den Windows- und Motif-Versionen der BAE-Benutzeroberfläche werden automatisch die Windows- bzw. Motif-spezifischen Popupmenüs zur Datei- bzw. Verzeichnisauswahl sowie Listboxes zur Elementnamensabfrage aktiviert.

Sofern möglich bzw. sinnvoll werden in den Datei- und Elementnamensabfragen Defaultnamen zur Auswahl angeboten (aktuelle Projektdatei, selektierte Bibliothek, aktueller Elementname, etc.). Mit den Dateiverwaltungsfunktion zum Löschen von Elementen können nur solche Dateielemente gelöscht werden, die nicht durch andere Elemente aus derselben DDB-Datei referenziert werden.

Automatische Sicherung der Designdaten

Der **Schaltplaneditor**, der **Layouteditor** und der **Neuronale Autorouter** sind mit einer Funktion zur automatischen Sicherung der aktuell bearbeiteten Designdaten ausgestattet. Die automatische Datensicherung kann mit der Funktion Autosave in den Parametermenüs aktiviert werden. Hierbei ist durch die Angabe eines nichtnegativen Integerwertes das Autosave Intervall für die automatische Datensicherung in Minuten zu spezifizieren. Die Angabe des Wertes 0 bzw. die Eingabe eines Bindestrichs (`-`) bewirkt dabei die Deaktivierung der automatischen Datensicherung. Ist die automatische Datensicherung aktiviert, dann wird das aktuell bearbeitete Element im spezifizierten Zeitintervall jeweils automatisch in einer Backupdatei gesichert. Um jedoch ein unbeabsichtiges Überschreiben des Inhalts der Backupdatei zu vermeiden, wird die automatische Datensicherung nur dann ausgeführt, wenn am aktuell geladenen Element während des eingestellten Autosave-Zeitintervalls tatsächlich Änderungen vorgenommen wurden. Der Name der Backupdatei wird aus der aktuellen Projektdatei abgeleitet und erhält die Dateinamenserweiterung **`.bak`**. Automatisch gesicherte Elemente können falls nötig z.B. mit der Funktion Ablegen auf Namen aus dem Menü Datei oder dem Utilityprogramm **COPYDDB** zurückkopiert werden.

Automatische Parametersicherung

Im **Bartels AutoEngineer** sind Funktionen zur automatischen Sicherung wichtiger Design- bzw. Bearbeitungsparameter implementiert. Gesichert werden dabei z.B. das Zeitintervall für die automatische Datensicherung, Name der aktuell geladenen Farbtabelle, Eingabe- und Hintergrundraster, Winkel- und Rasterfreigabe, Koordinatenanzeigemodus, Standardwinkel und Spiegelungsmodus für die Bauteilplatzierung, Standardtextgröße, Bibliothekszugriffspfade, Plotdateinamen, Standardleiterbahnbreiten, Mincon-Funktion, Airlinedarstellung, Platzierungsmatrix, Flächenfüllparameter, usw. Die Sicherung der Parameter erfolgt automatisch und dediziert mit dem bearbeiteten Layout bzw. Schaltplan oder allgemein für die bearbeitete Bibliothekshierarchie (Bauteil, Padstack, Pad, SCM-Symbol, etc.). Beim Laden eines Elements wird automatisch der entsprechende Parametersatz mitgeladen. Dadurch wird in komfortabler Weise eine spezifische Arbeitsumgebung zur Bearbeitung der selektierten Bibliothekshierarchie bzw. des selektierten Designobjekts aktiviert.

Arbeitsbereich

Bei der Erstellung eines neuen Elements werden Sie um die Angabe der Arbeitsbereichsdimensionen (d.h. der Elementgrenzen) gebeten. Der Arbeitsbereich ist nicht mit der Platinen- bzw. Bauteilumrandung zu verwechseln, sondern hat nur die Aufgabe, die Größe eines Elements definiert zu halten. Durch den Arbeitsbereich wird vermieden, dass sich Grafikelemente möglicherweise im Unendlichen versteckt halten. Da das System mit einer Fließpunkt-Datenbasis arbeitet, wäre dies sonst möglich. Der Arbeitsbereich entspricht somit der Papiergröße bei einer manuellen Zeichnung. Außerhalb des Arbeitsbereichs können sich keine Grafikelemente befinden. Der Arbeitsbereich kann mit den Funktionen Untere Elementgrenze und Obere Elementgrenze (im Menü Einstellungen) auch nachträglich vergrößert oder verkleinert werden. Des Weiteren hat der Arbeitsbereich auch Einfluss auf die Entscheidungen beim sogenannten Clipping. Kurz gesagt verschlechtert ein unnötig großer Arbeitsbereich die Geschwindigkeit des Bildaufbaus und bestimmter anderer Funktionen. Daher sollte der Arbeitsbereich nicht größer als notwendig gewählt werden, zumal er auch nachträglich erweitert werden kann.

Gruppenfunktionen

Die Gruppenfunktionen arbeiten nach dem Mengenprinzip, Sie können entweder einzelne Elemente oder auch alle in einem bestimmten Polygonzug befindlichen Elemente getrennt nach Elementtypen wahlweise zu der Gruppe hinzufügen (selektieren) oder auch wieder aus der Gruppe entfernen (deselektieren). Die in der Gruppe befindlichen Elemente werden mittels Highlight angezeigt. Genau diese Elemente sind dann auch von der gewählten Gruppenfunktion betroffen, so werden z.B. alle nach Ausführung der Gruppen-Auswahlfunktionen Gruppe ruecksetzen, Gruppe Polygon und Gruppe Einzelelement noch hell angezeigten Elemente dann mit Gruppe bewegen auch bewegt. Auch die Gruppenfunktionen können auf unterschiedlichen Hierarchieebenen (Schaltplanblatt, Symbol im **Schaltplaneditor**; Layout, Part im **Layouteditor**) verwendet werden.

Undo, Redo

Die Funktionen Undo und Redo aus dem Menü Bearbeiten ermöglichen das Rückgängigmachen und Wiederausführen der zuletzt ausgeführten Funktionen. Dabei macht Undo zunächst die Funktion selber rückgängig und Redo seinerseits wiederum die Funktion Undo, usw. Somit gewährleistet dieses wichtige Funktionspaar einerseits Datensicherheit und ermöglicht andererseits eine komfortable Überprüfung von Realisierungsalternativen.

Standardmässig unterstützt das System bis zu zwanzig Undo-Schritte. Im Setup des BAE-Hauptmenü kann die Anzahl der möglichen Undo-Schritte für **Schaltplaneditor** und **Layouteditor** getrennt wahlweise auf bis zu hundert Schritte erhöht werden.

1.2.4 Eingaben im Grafikarbeitsbereich

Eingaben im Grafikarbeitsbereich werden mit Hilfe des Fadenkreuzes und der Maus vorgenommen. Je nach Funktion wird hierbei zwischen der Auswahl eines bestehenden grafischen Elements (Pick) oder dem Platzieren eines neuen grafischen Elements (Place) unterschieden. Häufig folgt auch einer Pick- direkt eine Place-Funktion, z.B. beim Bewegen eines Bauteils.

Die Anwahl eines Elements beim Pick oder die Bestätigung der momentanen Fadenkreuzposition beim Place in der jeweils sinnrichtigen Folge, z.B. Ablegen des Bauteils oder Bestätigung der nächsten Ecke einer Leiterbahn wird durch das Betätigen der linken Maustaste ausgelöst. Beim Pick bricht das Betätigen der rechten Maustaste die Funktion ab, beim Place hingegen wird mit der rechten Maustaste ein in vielen Fällen recht umfangreiches Untermenü dargestellt, in dem Sie dann eine zu der aktuellen Funktion passende Operation (z.B. Drehen eines Bauteils) auslösen können.

Da viele funktionsspezifische Operationen über diese Untermenüs erreicht werden können, sollten Sie sich hiermit genau vertraut machen. So werden z.B. alle Drehungen, alle direkten Koordinateneingaben beim Positionieren, alle Kreisbogenoperationen und andere wichtige Operationen wie das Setzen von Durchkontaktierungen, das Umlegen eines Leiterbahnsegments auf eine andere Lage oder auch das Verbreitern eines Segments über diese Untermenüs angewählt.

Grundsätzlich kann jede Fläche und jeder Linienzug (siehe Menü Grafik im Schematic Editor bzw. Menü Flächen im Layouteditor) als Polygon eingegeben werden. Ein Polygon darf beliebig viele Kreisbogenbestandteile enthalten. Zunächst wird die erste Bogenecke grafisch eingegeben, dann die jeweilige Operation Bogen links oder Bogen rechts im Untermenü angewählt. Danach erfolgt die Eingabe des Mittelpunktes, der jeweils resultierende Kreis wird dynamisch am Fadenkreuz angezeigt. Zuletzt wird dann der Endpunkt eingegeben, wobei der resultierende Kreisbogen dargestellt wird.

Ein Vollkreis wird wie folgt erzeugt: Zunächst wählen Sie einen Eckpunkt an, selektieren dann Bogen links und definieren den Mittelpunkt. Die Funktion wird dann unmittelbar nach Eingabe des Mittelpunktes anstelle der Eingabe des Endpunktes mit Fertig beendet. Alternativ kann mit Hilfe der c-Taste zunächst der Kreismittelpunkt angewählt und dann mit einem zweiten Punkt der Radius des Kreises spezifiziert werden.

1.2.5 Spezielle Hinweise

In den vorhergehenden Abschnitten wurden die grundsätzlichen Bedienungsvorgänge erläutert. Sie können jetzt z.B. mit den von uns vorgegebenen Beispielen die einzelnen Systemfunktionen kennen lernen. Bevor Sie jedoch mit dem System eigene Projekte erstellen, sollten Sie sich mit dem Programm **BSETUP** und der Datenbank vertraut machen, da die darin getroffenen Vereinbarungen die spätere Ausgabe der Fertigungsdokumentation erheblich beeinflussen.

Wartezeiten bei komplexen Funktionen

Da manche Funktionen etwas Rechenzeit benötigen, können gerade bei komplexeren Funktionen wie z.B. Laden oder Gruppenfunktionen kurze Wartezeiten auftreten. In den Windows-Versionen der BAE-Software werden Wartezeiten dadurch angezeigt, dass der Kursor während der Wartezeit in ein Sanduhrsymbol umgewandelt wird. In den Standardbenutzeroberflächen des **Bartels AutoEngineer** wechselt dagegen die Farbe des Menübalkens während der Wartezeit (in rot) Farbe und wird erst nach erfolgreicher Beendigung der Funktion wieder zurückgesetzt (auf grün). Das gleiche gilt, wenn das System eine grafische Eingabe oder eine Tastatureingabe in der Eingabezeile erwartet. Sofern längere Wartezeiten auftreten können, wird der Fortgang der Berechnungen in der Mitteilungszeile gemeldet (Prozentanzeige). Bitte setzen Sie jedoch in keinem Fall den Computer zurück, nur weil möglicherweise eine kurze Wartezeit auftritt. Durch das Zurücksetzen können Sie einen erheblichen Datenverlust verursachen.

Datensicherung

Sofern Sie mit einem realen Projekt starten, möchten wir Sie auf die notwendige Datensicherung hinweisen. Wir erhalten immer wieder Anfragen bezüglich des Rettens von Designdaten auf fehlerhaften Festplatten. Dies ist aber in vielen Fällen nicht möglich, da nicht die einfachste Datensicherung betrieben wurde. Eine Festplatte kann z.B. durch Headcrash oder andere Defekte einen echten Datenverlust erleiden. Auch wir können dann kein Wunder bewirken, denn: Was weg ist, ist weg! Bitte sichern Sie daher täglich zumindest Ihre Projektdateien (DDB-Dateien) von Ihrer Festplatte auf Diskette oder Band! Beachten Sie in diesem Zusammenhang bitte auch die Möglichkeit der automatischen Datensicherung mit Hilfe der Funktion Autosave aus dem Menü Einstellungen (siehe oben).

Fertigungsfreigabe

Vor einer Fertigungsfreigabe sollten Sie in jedem Fall im **Layouteditor** mit der Funktion Batch-DRC (siehe Menü Utilities) einen Design Rule Check im Batchbetrieb starten und das Ergebnis im automatisch angezeigten Report betrachten und nur bei völliger Fehlerfreiheit des Layouts den CAM-Prozess starten. Auch beim CAM-Prozess dürfen keine Fehler gemeldet werden, da ansonsten das Ergebnis möglicherweise unbrauchbar ist. Des Weiteren ist es notwendig, vor Freigabe einer Serienfertigung einen Prototyp ausgiebig zu testen. Bei Beachtung dieser Hinweise werden Sie mit dem **Bartels AutoEngineer** auch gute Projektergebnisse erzielen.

1.3 Datenbank

1.3.1 Allgemeine Hinweise

Das Kernstück des **Bartels AutoEngineer** ist die den Bedürfnissen der Anwender optimal angepasste Datenbank. Da komplexe und vor allem unterschiedlich große Dateneinträge gespeichert und verarbeitet werden müssen, ist die Datenbank objektorientiert aufgebaut. Schneller Zugriff auf die gewünschten Informationen durch einen speziell hierfür entwickelten Suchalgorithmus vermeidet überflüssige Wartezeiten am Bildschirm. Ein ausgeklügeltes Programm zur Bibliotheksverwaltung ermöglicht es, mehrere Normbibliotheken parallel zu halten. Der Anwender kann also besondere Bibliotheken für einzelne Projekte oder Jobs definieren, sie mit dem Projekt abspeichern und - sofern gewünscht - diese Projektbibliotheken auch in eine Hauptbibliothek des Systems schreiben.

Objektklassen, Hierarchie

Die einzelnen Datenbankeinträge sind nach Klassen sortiert. Innerhalb einer Klasse wird ein Element über seinen Namen eindeutig identifiziert. Jedes Element enthält die Daten, durch die es unmittelbar beschrieben wird (z.B. grafische Elemente, Texte, Anschlusspositionen und Pinbezeichnungen bei Stromlauf- bzw. Layoutsymbolen). Darüber hinaus kann jedes Element einer Datenbankklasse Verweise auf andere Elemente aus der in der Hierarchieebene darunterliegenden Klasse beinhalten. So enthält z.B. ein Layout Verweise auf die in ihm verwendeten Bauteile, die Bauteile wiederum Verweise auf die darin enthaltenen Padstacks, und diese wiederum Verweise auf die Pads, aus denen sie aufgebaut sind. Alle diese Verweise sind der Datenbank bekannt. So werden z.B. beim Laden oder Kopieren eines Bauteils alle dazugehörigen Padstack-Elemente und mit diesen wiederum alle zugehörigen Pad-Elemente gegebenenfalls mit übernommen. All dies geschieht vollautomatisch und transparent.

Homogenität

Alle Datenbankklassen unterliegen den allgemeinen Dateiverwaltungsfunktionen. D.h., der **Schaltplaneditor** ist zum Erstellen und Verändern von Stromlaufplänen, Stromlauf-, Label- und Pinsymbolen gleichermaßen geeignet. Gleiches gilt natürlich auch für den **Layouteditor** in Bezug auf die Erstellung bzw. Änderung von Layouts, Bauteilen, Padstacks und Pads. Die Funktionen werden jeweils automatisch entsprechend angepasst, d.h. die jeweiligen Menüs sind sowohl im Schaltplan-Paket als auch im Layout auf allen Hierarchieebenen weitgehend identisch. Die dadurch gegebenen weit reichenden Analogien bei der Bearbeitung von Objekten jeglicher Art gewährleisten eine leichte Erlernbarkeit des Systems.

Dateiformat

Die verschiedenen Programme des **Bartels AutoEngineer** arbeiten alle mit demselben Datenbank- bzw. Dateiformat. Dieses Format nennen wir Design DataBase- oder kurz DDB-Format. Die File-Extension des Dateityps, auf die dieses Format abgebildet wird, lautet grundsätzlich `.ddb`. Die Projektdateien des **Bartels AutoEngineer** enthalten alle jeweils für ein Projekt relevanten Informationen. So können sämtliche Designdaten für ein beliebig großes Projekt (Stromlaufplan, Leiterplatten-Layout, Netzliste, Bibliothek, Steuerdaten) in einer einzigen DDB-Datei gespeichert werden, die damit das komplette Projekt beschreibt. Auch die Bibliotheksdateien (mit Stromlaufsymbolen, Layoutsymbolen, Logikbibliothek, usw.) sind im DDB-Format erstellt.

Datenkonsistenz

Ein Element wird während eines Ladevorgangs grundsätzlich dynamisch aufgebaut. So wird beim Laden eines Layouts zunächst das Layoutelement in den Speicher gelesen und dann die zugehörigen Bauteil-Elemente aus der selben Projektdatei; diese wiederum laden die benötigten Padstacks, usw. Dieser dynamische Ladevorgang setzt natürlich voraus, dass z.B. beim erstmaligen Platzieren eines neuen Bauteils alle benötigten Elemente wie etwa das Bauteil selbst, die Padstack- oder auch Pad-Elemente in der Projektdatei vorhanden sein müssen. Dies wird vom System automatisch geprüft. Sofern die Elemente nicht vorhanden sind, werden genau diese Elemente aus der gewählten Symbol- bzw. Bauteilbibliothek in die aktuell bearbeitete Projektdatei geladen. Somit wird automatisch eine projektspezifische Symbol- und Bauteilbibliothek innerhalb der bearbeiteten Projetdatei erstellt. Durch die Speicherung jobspezifischer Bibliotheken innerhalb der Projektdateien ergeben sich immense Vorteile z.B. hinsichtlich der Archivierung von Projekten oder der geringen Abhängigkeit von der Verfügbarkeit von Master-Bibliotheken. Auf der anderen Seite sichert dieses Verfahren auch die Datenkonsistenz bei etwaigen Änderungen in der Bibliothek. Mit den Dateiverwaltungsfunktionen zum Löschen von Elementen können nur solche Dateielemente gelöscht werden, die nicht durch andere Elemente aus derselben DDB-Datei referenziert werden. Die Funktionen zur Verwaltung der Datenbank sorgen dafür, dass Änderungen nur kontrolliert übernommen werden. So ist es z.B. möglich, mit der Funktion Update Bibliothek (im Menü Datei) in kürzester Zeit eine komplette Aktualisierung der jobspezifischen Bibliothek eines Projektes herbeizuführen. Ein typischer Anwendungsfall hierfür ist die Angleichung an eine Master-Bibliothek bei einem Redesign eines zuvor archivierten Projektes.

Kontrollierte Fertigungsanpassung

Eine weiterer Vorteil des Datenbankkonzepts besteht in der Möglichkeit der kontrollierten Anpassung an die Fertigung. So ist es möglich, projektbezogene Änderungen durchzuführen, d.h. spezielle Bauteile, Padformen, usw. zu erstellen, und diese gezielt für bestimmte Projekte einzusetzen. Ein Beispiel hierfür wäre die Erstellung eines Technologiebauteils, welches alle Padstacks einer Technologie, z.B. alle SMD-Pads beinhaltet. Dieses Bauteil kommt dann nicht im Layout vor, sondern wird als Element für einen manuellen Kopiervorgang verwendet. Die Padstacks und Pads wiederum können in verschiedenen Technologiebibliotheken verschiedene Formen und Größen haben. Bei Element ersetzen oder Update Bibliothek mit der Projektdatei und dem Technologiebauteil als Zielangabe und einer dieser Technologiebibliotheken als Quellangabe werden dann genau die mit dem Technologiebauteil gewählten Elemente aus der Technologiebibliothek in das Projekt kopiert.

Erstellung von Bibliothekselementen

Die Erstellung komplett neuer Bibliothekselemente geschieht im Stromlauf- wie im **Layouteditor** grundsätzlich mit der Funktion Neues Element aus dem Menü Datei. Nach der Spezifikation der Hierarchieebene, in der das Element zu erstellen ist, wird der Anwender nach dem Namen der DDB-Datei, in der das Element abgelegt werden soll, dem Namen des zu erstellenden Elements selbst, sowie den Elementgrenzen gefragt. Anschließend erfolgt die eigentliche Definition des Elements durch Laden und Positionieren von Elementen der darunterliegenden Hierarchieebene und durch Einbringung zusätzlicher (in der aktuellen Hierarchieebene zulässiger) Daten (Zeichnungsinformation, Texte, usw.).

SQL-Funktionen, Relationale Datenbanken

Über die **Bartels User Language** werden SQL (Structured Query Language)-Funktionen zur Verwaltung Relationaler Datenbanken angeboten. Damit stehen dem Anwender Software-Tools zur Programmierung von Datenbankmanagement-Systemen zur Verfügung. Mit den SQL-Funktionen ist es möglich, Datenbanksysteme zur Adress- und Projektverwaltung, zum Projektmanagement, zur Versionsverwaltung, zur Produktionsplanung und -steuerung (PPS), zur Verwaltung von Lieferanten- und Kundenverzeichnissen, usw. zu realisieren. Eine detaillierte Beschreibung der **Bartels User Language** und der darin integrierten SQL-Funktionen finden Sie im Bartels User Language Programmierhandbuch.

1.3.2 Datenbankhierarchie im Stromlauf

Abbildung 1-4 zeigt beispielhaft das Schema der Datenbankhierarchie im Stromlaufpaket des **Bartels AutoEngineer**.

Die Planebene ist die oberste Hierarchieebene im Stromlauf. Auf dieser Ebene erfolgt die Eingabe des Stromlaufplans. D.h., hier werden Stromlaufblätter definiert, Symbole platziert und über Verbindungen, Busse und Labels eine Netzliste definiert. Zusätzlich kann Grafik und Text zur Dokumentation mit eingebracht werden.

In der Symbolebene werden die Stromlaufsymbole, also die Schaltzeichen erstellt (und in Form von Stromlauf-Bibliothekselementen in einer Schaltzeichenbibliothek abgelegt). Im wesentlichen werden in dieser Ebene durch Selektieren und Platzieren von Elementen aus der darunterliegenden Pinebene die logischen Anschlüsse des Schaltzeichens in Form und Position festgelegt. Daneben lassen sich hier Grafik und Texte (z.B. Symbol-Outline und Referenz für den Bauteilnamen) einbringen.

In der Labelebene werden Spezialsymbole zur Definition von Netznamen erstellt. Mit Hilfe dieser Symbole können im Stromlauf die Verbindungen explizit auf definierte Signale bzw. Signalpegel (auch stromlaufblattübergreifend) gelegt werden.

In der Markerebene werden über die Definition eines Kontaktbereiches Pinsymbole erstellt. Diese Pinsymbole lassen sich auf Symbol- oder Labelebene platzieren und legen somit die Form und Position der entsprechenden Anschlüsse fest. Neben den normalen Pinsymbolen lässt sich in dieser Hierarchieebene auch ein spezielles Markersymbol (mit einer normalen Grafikfläche anstelle des Kontaktbereiches) definieren, welches nach einer entsprechenden Selektion auf Planebene zur Kenntlichmachung von T-förmigen Verbindungsstücken dient.

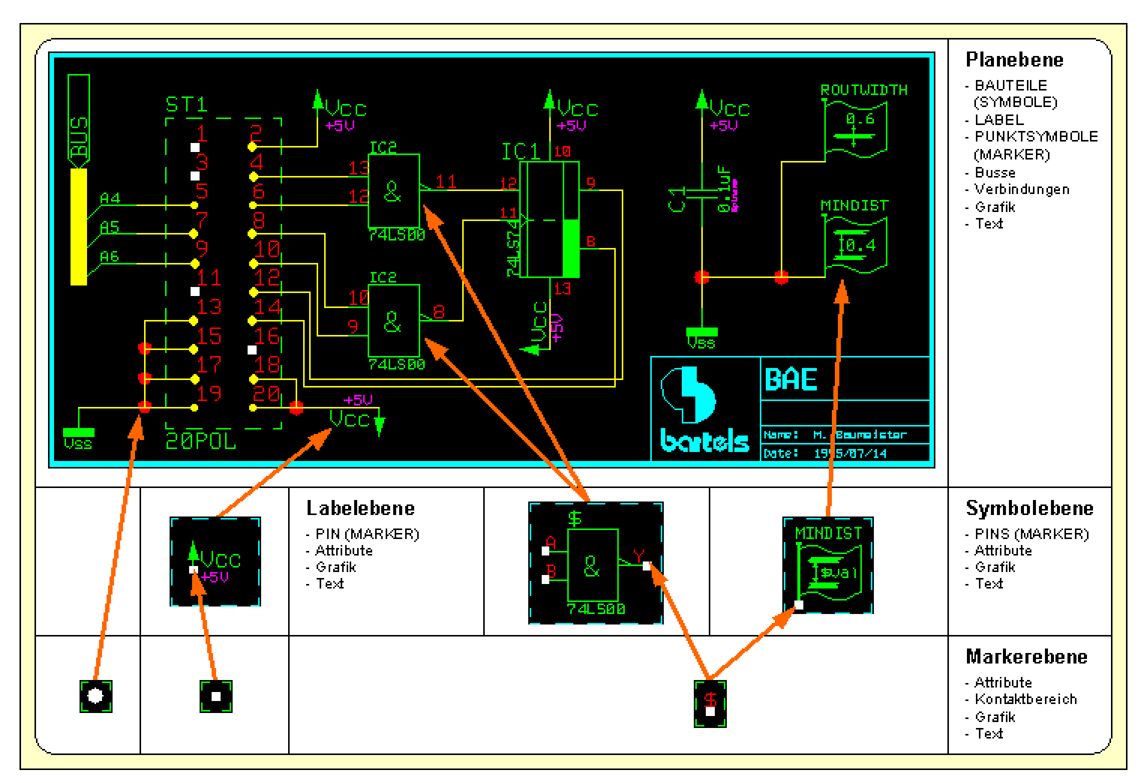


Abbildung 1-4: Datenbankhierarchie im Stromlauf

1.3.3 Datenbankhierarchie im Layout

Abbildung 1-5 zeigt beispielhaft das Schema der Datenbankhierarchie im Layoutsystem des **Bartels AutoEngineer**.

Die Layoutebene ist die höchste Hierarchieebene im Layout. Hier wird das Leiterplatten-Layout bearbeitet. D.h., die Leiterkartenkontur wird festgelegt, die Bauteile (aus der darunterliegenden Bauteilebene) platziert, Sperr-, Potential- und Kupferflächen definiert, die Leiterbahnen verlegt, und schließlich die Ausgabedaten für die Fertigung generiert.

In der Bauteil- bzw. Partebene werden die Layoutsymbole, also die Gehäusebauformen definiert (und in Form von Layoutbibliothekselementen in einer Gehäusebibliothek abgelegt). Im wesentlichen werden in dieser Ebene durch Selektieren und Platzieren von Elementen aus der darunterliegenden Padstackebene die Bauteilanschlüsse in Form und Position festgelegt. Daneben lassen sich hier Leiterbahnen und Vias (z.B. für gedruckte Spulen) platzieren, Sperr- oder Kupferflächen definieren sowie Zeichnungsinformationen und Texte (z.B. Bauteilumriss und Referenz für den Bauteilnamen auf dem Bestückungsplan) aufbringen.

In der Padstackebene werden durch selektieren und positionieren von Padsymbolen die Pinsymbole erstellt, die lagenbezogen aus verschiedenen Padformen aufgebaut sein können und für gebohrte Pins eine entsprechende Bohrung enthalten. Auch Texte (z.B. Referenz für den Pinnamen), Zeichnungsinformation (z.B. für den Bohrplan) und Sperrflächen (z.B. zur Bestimmung der Anschlussart durch den **Autorouter**) können hier eingebracht werden. Die in dieser Ebene generierten Padstacks lassen sich auf Bauteilebene platzieren und definieren somit in Form und Position die Bauteilanschlüsse.

In der Padebene werden durch die Definition von Kupferflächen Anschlussformen festgelegt. Die in dieser Ebene generierten Pads werden dann in der Padstackebene über die verschiedenen Lagen zu einem Stapel (Padstack) zusammengefasst und bilden die eigentliche Bauteil-Anschlussdefinition.

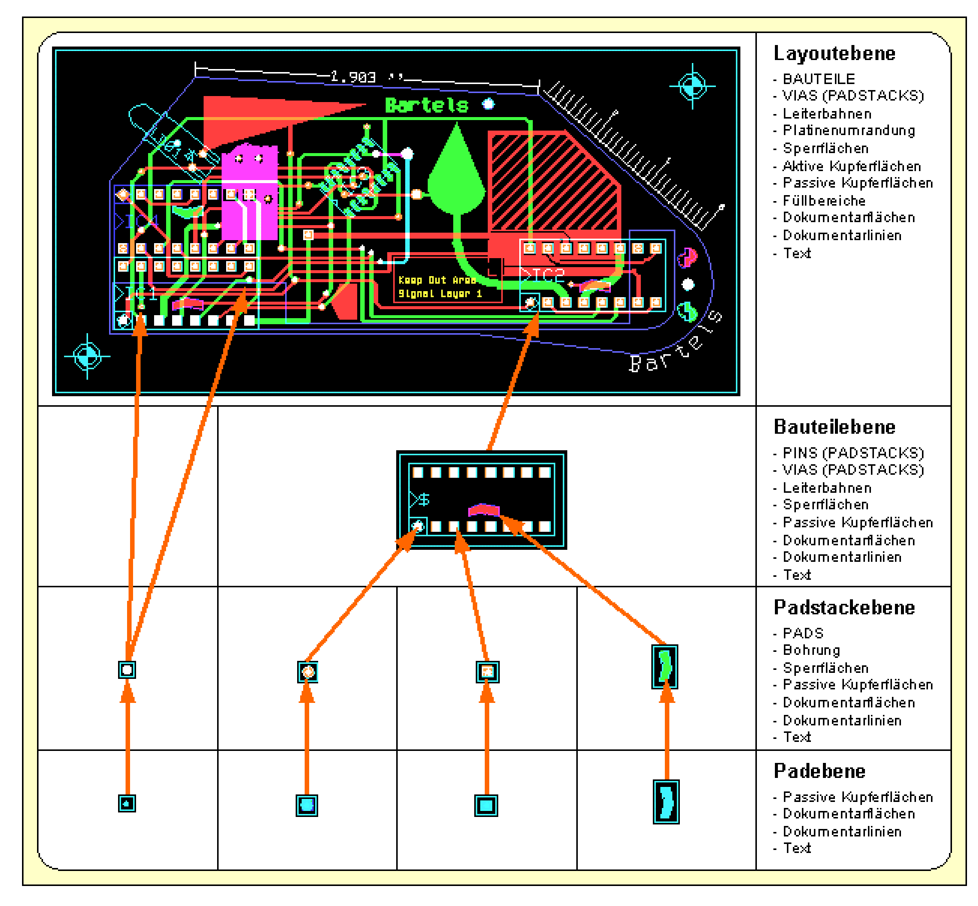


Abbildung 1-5: Datenbankhierarchie im Layout

1.3.4 Logische Bibliothek

Die logische Bibliothek stellt das Bindeglied zwischen Stromlauf- und Layoutbibliothek dar. In dieser logischen Bibliothek sind Informationen über die Zuordnung der Stromlauf- zu den Layoutsymbolen (mit Gatterdefinitionen und Pin-Mapping), Pin-, Gatter- und Pingruppen-Vertauschbarkeit, feste Anschlüsse zur Stromversorgung, usw., enthalten. Alle diese Definitionen können mit Hilfe eines Editors in einer ASCII-Datei eingetragen (siehe hierzu Beschreibung für das Programm **LOGLIB** in Kapitel 7.11 dieses Handbuchs) und anschließend mit dem Programm **LOGLIB** in eine DDB-Datei eingespielt werden. Die Informationen aus der logischen Bibliothek benötigt der **Packager**, um eine (mit dem **Schaltplaneditor** erzeugte) ungepackte, logische Netzliste in eine gepackte, physikalische (d.h., eine für das Layoutsystem verständliche) Netzliste zu transferieren (siehe hierzu auch die Beschreibung zum **Packager** in Kapitel 3.2). Während eines **Packager**-Laufs müssen die in der logischen Bibliothek definierten Einträge zusätzlich auch gegen die Einträge in der Layoutbibliothek geprüft werden. Da der **Packager** nur eine Bibliotheksdatei auswerten kann, setzt dies voraus, dass sowohl die logische als auch die Layoutbibliothek in einer einzigen Bibliotheksdatei gespeichert sein müssen.

Abbildung 1-6 zeigt ein Beispiel für eine Bauteildefinition wie sie entsprechend Datenblatt in der logischen Bibliotheksdatei vorzunehmen ist.

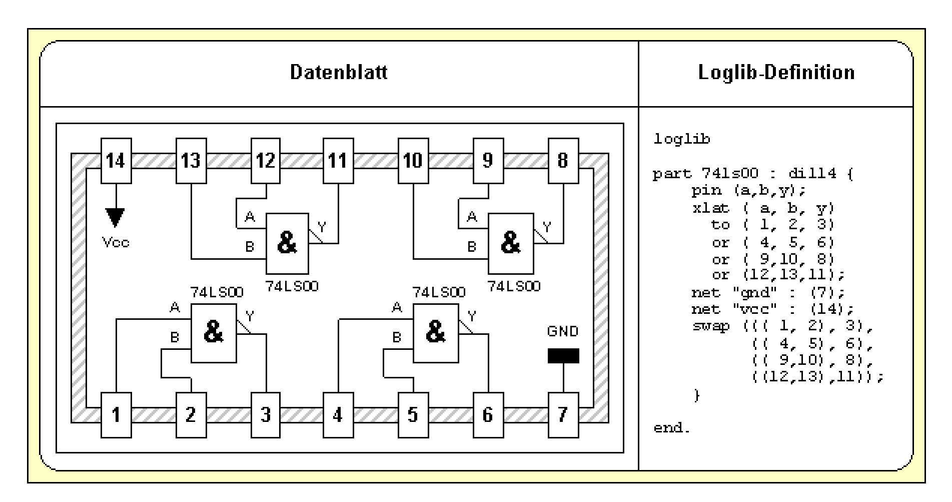


Abbildung 1-6: Loglib-Bauteildefinition entsprechend Datenblatt

Bei der Erstellung eines komplett neuen Bauteiles empfehlen wir grundsätzlich folgende Vorgehensweise:

- Erstellen des Stromlaufsymbols und Ablegen dieses Schaltzeichens in einer Stromlaufbibliothek
- Erstellen des Layoutsymbols und Ablegen dieses Symbols in der zentralen Layoutbibliothek (falls die entsprechende Gehäusebauform nicht schon existiert)
- Erstellen der ASCII-**LOGLIB**-Datei mit den Definitionen für die logische Bibliothek und
- Einspielen der **LOGLIB**-Datei in die zentrale Layoutbibliothek mit Hilfe des Programms loglib

Nach Ausführung obiger Arbeitschritte kann das neu definierte Schaltzeichen bei der Erstellung eines Stromlaufs verwendet werden, der **Packager** ist in der Lage dieses Schaltzeichen in das richtige Gehäuse zu packen, und die Gehäusebauform ist für die Platzierung im Layout verfügbar. Weniger erfahrene Anwender sollten neue Bibliothekselemente zunächst in einer Testdatei erstellen und einen **Packager**-Testlauf durchführen, bevor die neue Bauteildefinition für reale Projekte freigegeben wird. Dadurch lässt sich die Übernahme fehlerhafter Definitionen in Bibliotheks- bzw. Projektdateien verhindern.

Kapitel 2
Schaltungsentwurf / CAE

Dieses Kapitel erläutert die Handhabung des Programm-Moduls **Schematic Editor** für den Entwurf von Schaltplänen. Hierbei wird der Leser in einer logischen Abfolge durch die Erstellung von Bibliothekselementen sowie die Bearbeitung von Stromlaufplänen geführt. Der in diesem Kapitel erstellte Schaltplan wird in den nachfolgenden Kapiteln einer weiteren Bearbeitung unterzogen. Es wird daher empfohlen, dieses Kapitel Schritt für Schritt und ohne Auslassung irgendwelcher Abschnitte durchzuarbeiten, um einen vollständigen Überblick über den Funktionsumfang des **Schematic Editors** zu gewinnen. Sobald ein spezielles Kommando angewandt bzw. dessen Benutzung erläutert wurde, ist davon auszugehen, dass der Leser dieses Kommando verstanden hat und es bei Bedarf ohne nähere Erläuterungen wieder ausführen kann. Nachfolgende Instruktionen zu dem betreffenden Kommando sind dann weniger ausführlich, um das Lesen zu vereinfachen und den Lernprozess zu beschleunigen.

Inhalt

Tabellen

Abbildungen

2.1 Allgemeine Hinweise

Das Schaltplanpaket des **Bartels AutoEngineer** besteht im Wesentlichen aus einem grafisch-interaktiven **Schaltplaneditor** mit integriertem Stromlaufsymboleditor und integrierter Backannotation sowie dem Programm-Modul **Packager** zur Umwandlung von logischen in physikalische Netzlisten ("Forward Annotation"). Die nachfolgenden Abschnitte dieses Benutzerhandbuchs enthalten eine detaillierte Beschreibung des **Schaltplaneditors** zur Erstellung und Bearbeitung von Stromlaufsymbolen und Schaltplänen.

2.1.1 Komponenten und Leistungsmerkmale

Schaltplaneditor

Wer mit einem System zur Schaltplanerstellung arbeitet, erwartet vor allem, dass seine Netzliste zuverlässig exakt mit dem Schaltbild übereinstimmt. Im **Schaltplaneditor** des **Bartels AutoEngineer** wird deshalb die Herstellung einer Verbindung, beispielsweise der Anschluss eines Pins an eine Leitung, sofort vom Computer farblich quittiert. Auch T-Stücke werden vom System selbständig erkannt und durch Anschlusspunkte markiert. Diese Zuverlässigkeit wird dadurch erreicht, dass die Netzliste im Hintergrund "inkremental" aktualisiert, das heißt nur entsprechend der gerade durchgeführten Änderung neu berechnet wird. Hierzu wird eine Änderungsliste geführt, die auch die zwanzigstufige Undo/Redo-Funktion steuert.

Eine weitere Anforderung an ein CAE-System ist die völlige Freiheit bei der Erstellung der Schaltplansymbole. Mit dem **Bartels AutoEngineer** kann der Anwender am Bildschirm eigene Symbole "zeichnen" und ist so nicht auf eine vorgegebene Norm angewiesen. Einmal entwickelt und gespeichert, dreht und spiegelt das System die Symbole entsprechend den Vorgaben selbsttätig.

Invertierte Signale werden mit einem Strich über dem Text gekennzeichnet, der - auch beim Versetzen des Textes - fest mit diesem verbunden bleibt. Beim Aufrufen von Signalnamen werden - vom Anwender konfigurierbar - automatisch die richtigen Labels verwendet, z.B. das in der Bibliothek vordefinierte Massesymbol für Masseverbindungen.

Selbstverständlich sind dem System auch Busse und Bustaps bekannt, wobei nicht nur numerische, sondern beliebige Namen für die Bussignale vergeben werden können. Durch die leistungsfähige inkrementale Netzlistenerzeugung erkennt das System außerdem auch über Busse und Labels entstandene Verbindungen bereits während der Eingabe. Im **Bartels AutoEngineer** können Busse jedoch nicht nur auf Schaltplanebene sondern bereits auf Symbolebene in der SCM-Bibliothek definiert werden. Dieses flexible Konzept der Bussynthese bei der Symbolgestaltung unterstützt den Anwender optimal beim Entwurf beliebig komplexer Bauteile.

Das System unterstützt den Anwender auch bei der Erzeugung von hierarchischen Schaltplänen. D.h., es ist möglich, Schaltpläne als Blockschaltbilder zu definieren und diese auf anderen Stromläufen als Block zu referenzieren. Selbstverständlich erfolgt hierbei ebenfalls eine saubere Verwaltung der Netzliste mit der logisch richtigen Unterscheidung zwischen globalen und lokalen Netzen.

Irren ist menschlich - und gerade der Entwickler muss die Möglichkeit haben, verschiedene Lösungsansätze auszuprobieren und durchzuspielen und auf Wunsch wieder an den Ausgangspunkt seiner Überlegungen zurückzukehren. Deshalb können im Bartels-System die letzten zwanzig Arbeitsschritte mit der Undo-Funktion rückgängig gemacht werden. Entscheidet sich der Anwender dann trotzdem für den zuerst eingeschlagenen Weg, kann er den Undo-Befehl per Redo aufheben.

Durch die Einbindung der **Bartels User Language** in das Schaltplan-Paket hat der Anwender die Möglichkeit, eigene Menüfunktionen (Makros), Postprozessoren, Test- und Editierfunktionen, usw. zu implementieren, die er wahlweise explizit (über eine spezielle Menüfunktion) oder implizit (über Tastatur oder ereignisgesteuert) aktivieren kann.

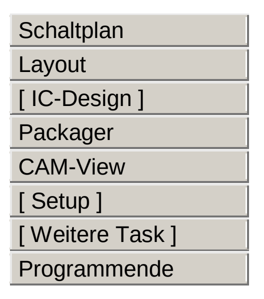
Das Stromlaufpaket des **Bartels AutoEngineer** bietet eine Reihe mächtiger zusätzlicher bzw. spezieller Features wie z.B. Gruppenfunktionen, beliebige Attributdefinitionen, Attributwertzuweisungen an Bauteile bzw. Netze, automatische Bauteilbenennung mit definierbaren Namensmustern, virtuelle Symbole für Firmenlogos oder Schriftfelder, Netz-Highlight, automatisches Re-connect beim Bewegen von Symbolen, usw. Netzlistenänderungen im Layout wie z.B. Pin/Gate-Swaps oder die Umbenennung von Bauteilen werden mit Hilfe der Backannotation automatisch in den Schaltplan zurückgemeldet.

2.1.2 Programmaufruf

Der Aufruf des **Bartels AutoEngineer** sollte grundsätzlich aus dem Verzeichnis erfolgen, in welchem die zu bearbeitenden Projektdateien abgelegt bzw. abzulegen sind. Wechseln Sie also zunächst in Ihr Projektverzeichnis (zur Abarbeitung der in diesem Handbuch aufgeführten Beispiele ist es zweckmäßig, in das bei der Installation des **Bartels AutoEngineer** angelegte BAE-Jobs-Directory zu wechseln). Der Aufruf des Stromlauf-Editors erfolgt aus der Shell des **Bartels AutoEngineer**. Starten Sie diese von Betriebssystemebene aus mit folgendem Befehl:



Der **AutoEngineer** zeigt auf dem Schirm das Bartels-Logo sowie folgendes Menü (die Funktion Setup ist nur unter Windows bzw. Motif verfügbar; die Menüpunkte IC-Design und Weitere Task sind nur in speziellen Softwarekonfigurationen wie etwa in **BAE HighEnd** oder **BAE IC Design** verfügbar):



Wählen Sie den Menüpunkt Schaltplan mit der Maus an, und bestätigen Sie Ihre Wahl durch Drücken der linken Maustaste:



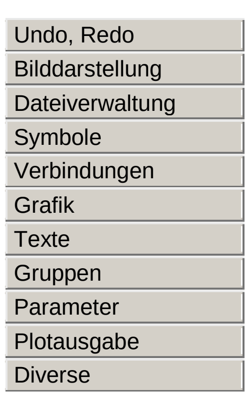
Nun wird der **Schaltplaneditor** des **AutoEngineers** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

2.1.3 Hauptmenü

Nach dem Aufruf des Stromlauf-Editors befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden des Schematic-Moduls ist das Menü Dateiverwaltung aktiviert, und der grüne Menübalken steht auf Laden.

Unter Windows und Motif kann anstelle der Standard- bzw. Seitenmenükonfiguration wahlweise auch ein Benutzerinterface mit Pulldownmenüs aktiviert werden. Hierzu ist mit Hilfe des Utilityprogramms **BSETUP** das Kommando **WINMENUMODE** mit der Option **PULLDOWN** in das Setup der BAE-Software einzuspielen (siehe hierzu auch Kapitel 7.2). Bei der Verwendung von Pulldownmenüs ist das Hauptmenü als horizontal ausgerichtete Menüleiste am oberen Ende der Benutzerschnittstelle angeordnet.

Das Hauptmenü ist während der Dauer der Stromlaufbearbeitung ständig verfügbar und ermöglicht die Aktivierung der folgenden Menüs:



Undo, Redo

Im Menü Undo, Redo finden Sie die Undo-Funktion, mit der die letzten zwanzig Arbeitsschritte rückgängig gemacht werden können. Mit der Redo-Funktion kann der Undo-Befehl wieder aufgehoben werden. Sie sollten diese wichtigen Funktionen unbedingt an einigen Stellen in den nachfolgenden Beispielen ausprobieren, um ein Gefühl für die Mächtigkeit dieser Kommandos zu bekommen.

Ansicht, Bilddarstellung

Im Menü Ansicht bzw. Bilddarstellungs, das Sie außer durch Selektion im Hauptmenü auch immer über die mittlere Maustaste erreichen, können Sie Zoomfunktionen aktivieren, das Eingabe- bzw. Hintergrundraster definieren, oder die Farbtabelle einstellen. Daneben existieren hier nützliche Hilfsfunktionen z.B. zur Bauteilsuche oder zur Netzanzeige.

Dateiverwaltung

Über das Menü Dateiverwaltung können Elemente neu generiert, geladen, gespeichert, kopiert, ersetzt oder gelöscht werden. Außerdem können von hier aus Farbtabellen geladen oder gespeichert werden, und es sind in diesem Menü auch wichtige Datenbank-Verwaltungsfunktionen (Auflisten Dateiinhalt, Update Bibliothek) enthalten.

Symbole

Das Menü Symbole dient dazu, Schaltzeichen, Labels oder Modulports in den Schaltplan zu laden, diese zu bewegen oder wieder zu löschen. Außerdem stehen hier Funktionen zur Zuweisung von Attributwerten an Bauteile sowie zur Anzeige der in der Logischen Bauteilbibliothek definierten Symbollogik zur Verfügung. Auf Symbol- bzw. Labelebene können über das Menü Symbole Pins (d.h. Markersymbole) platziert, bewegt und gelöscht sowie Namensmuster für die automatische Bauteilbenennung definiert werden.

Verbindungen

Das Menü Verbindungen dient dazu, Verbindungen oder Busse zu generieren, umzuverlegen oder wieder zu löschen.

Grafik

Das Menü Grafik enthält Funktionen, um Grafik (Linien oder Flächen) zu erzeugen, diese zu verändern, zu bewegen, zu kopieren oder wieder zu löschen.

Texte

Das Menü Texte dient dazu, Texte einzugeben, zu bewegen, zu verändern, oder wieder zu löschen.

Gruppen

Im Menü Gruppen werden Funktionen angeboten, mit deren Hilfe selektierbare Teile des gesamten Schaltplans in Gruppen zusammengefasst und dann bewegt, kopiert oder gelöscht werden können.

Parameter

Das Menü Parameter enthält Funktionen zur Selektion der Bibliothek, zum Setzen des Nullpunktes bzw. der Elementgrenzen, zur Selektion des Pinsymbols und des Verbindungspunkt-Markers, zur Festlegung der Schaltplanhierarchie, sowie zur Aktivierung der automatischen Datensicherung.

Plotausgabe

Im Menü Plotausgabe sind die Funktionen zur Erstellung von HP-GL-, HP-Laser- und Postscript-Ausgabedaten enthalten.

Diverse

Im Menü Diverse kann ein Rücksprung in die Shell des **Bartels AutoEngineer** oder der Programmabbruch veranlasst werden. Außerdem besteht von hier aus die Möglichkeit des expliziten **User Language**-Programmaufrufs.

2.1.4 Modifizierte Benutzeroberfläche

Menübelegung und Tastaturprogrammierung

Einige der mit der BAE-Software installierten **User Language**-Programme definieren implizite **User Language**-Programmaufrufe über die eine weit reichend modifizierte Benutzeroberfläche mit einer Vielzahl von Zusatzfunktionen (Startups, Toolbars, Menübelegung, Tastaturprogrammierung) aktiviert wird. Das **User Language**-Startupprogramm **BAE_ST** wird automatisch beim Aufruf des **Schematic Editors** gestartet. **BAE_ST** ruft seinerseits das **User Language**-Programm **UIFSETUP** auf, welches eine vordefinierte Menü- und Tastaturbelegung im **Schaltplaneditor** aktiviert. Änderungen bzw. Anpassungen der Menü- und Tastaturbelegung können *zentral* in der Quellcodedatei von **UIFSETUP** vorgenommen werden. Die aktuelle Tastaturbelegung kann mit dem **User Language**-Programm **HLPKEYS** angezeigt werden. Der Aufruf von **HLPKEYS** ist über die Funktion Tastaturbelegung aus dem Menü Hilfe möglich, sofern die vordefinierte Menübelegung aus **UIFSETUP** aktiviert ist. Mit dem **User Language**-Programm **UIFDUMP** kann die in der aktuellen Interpreterumgebung definierte Menü- und Tastaturbelegung in Form eines Reports angezeigt bzw. auf eine Datei ausgegeben werden. Mit dem **User Language**-Programm **UIFRESET** lässt sich die komplette Menü- und Tastaturbelegung zurücksetzen. **UIFSETUP**, **UIFDUMP** und **UIFRESET** sind auch über das Menü des **User Language**-Programms **KEYPROG** aufrufbar, welches zudem komfortable Funktionen zur Online-Tastaturprogrammierung sowie zur Verwaltung von Hilfstexten für **User Language**-Programme zur Verfügung stellt.

Kontextmenüs im Grafikarbeitsbereich

Bei Betätigung der linken Maustaste im Grafikarbeitsbereich wird ein kontextsensitives Menü mit spezifischen Funktionen zur Bearbeitung des an der aktuellen Mausposition platzierten Objekts aktiviert, wenn nicht bereits eine andere Menüfunktion aktiv ist. Ist kein Element geladen, dann werden die Dateiverwaltungsfunktionen Element laden bzw. Neues Element angeboten. Dieses Feature ist über einen automatisierten Aufruf des **User Language**-Programms **SCM_MS** implementiert.

Kaskadierende Pulldownmenüs unter Windows/Motif

Die Windows- und Motifversionen des **Schematic Editors** ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Pulldownmenüs der Windows- und Motifversionen des **Schematic Editors** werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs ausgestattet. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

Dialoge für Parametereinstellungen unter Windows/Motif

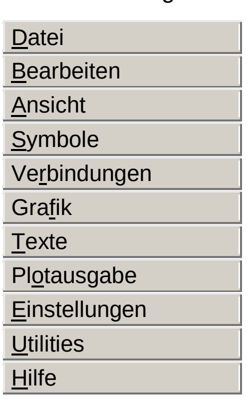
In den Windows- und Motifversionen des **Schematic Editors** sind die folgenden Dialoge für Parametereinstellungen implementiert:

- Einstellungen - Einstellungen: Allgemeine SCM-Parameter
- Ansicht - Einstellungen: Bilddarstellungsparameter
- Plotausgabe - Einstellungen: Plotparameter

In den Pulldownmenükonfigurationen werden die Standardfunktionen für Parametereinstellungen über das **User Language**-Programm **UIFSETUP** durch die obigen Menüfunktionen zum Aufruf der entsprechenden Dialoge ersetzt.

Pulldownmenükonfiguration unter Windows/Motif

Bei der Verwendung von Pulldownmenüs unter Windows und Motif wird über das **User Language**-Programm **UIFSETUP** eine an Windows angepasste Menüanordnung mit zum Teil geänderten Funktionsbezeichnungen und einer Vielzahl von Zusatzfunktionen konfiguriert. Das Hauptmenü des **Schematic Editors** wird dabei wie folgt aufgebaut:



Das Menü Hilfe enthält die beiden Funktionen Referenzhandbuch und Hilfe zu. für den Zugriff auf das im Windows-Help-Format verfügbare Referenzhandbuch zum **Schaltplaneditor**. Hilfe zu lädt dabei direkt die Referenzhandbuchseite eines selektierbaren Menüpunkts oder Benutzeroberflächenelements.

2.1.5 Grundsätzliches zur Bedienung

Automatische Parametersicherung

Im **Schaltplaneditor** ist eine Funktion zur automatischen Sicherung wichtiger Design- und Bearbeitungsparameter implementiert. Bei Aktivierung der Funktion zur Sicherung des aktuell geladenen Elements werden die folgenden Parameter automatisch in der aktuell bearbeiteten Designdatei gespeichert:

- Zeitintervall für automatische Datensicherung
- Name der aktuell geladenen SCM-Farbtabelle
- Eingaberaster
- Hintergrundraster
- Raster- und Winkelfreigabe
- Koordinatenanzeigemodus
- Standardwinkel für Symbol- und Labelplatzierung
- Spiegelungsmodus für Symbol- und Labelplatzierung
- Signalroutingmodus Ein/Aus für Symbol- und Labelplatzierung
- Spiegelungsmodus für Bustapplatzierung
- Standardtextgröße
- Bibliothekszugriffspfad
- Pfadname Logische Bibliothek
- Plotdateiname

Die Elementnamen der zu sichernden Parametersätze werden vom aktuell bearbeiteten SCM-Element abgeleitet. Planspezifische Parametersätze erhalten den Namen **`[plan]`**, symbolspezifische Parametersätze den Namen **`[symbol]`**, labelspezifische Parametersätze den Namen **`[label]`**, markerspezifische Parametersätze den Namen **`[marker]`**. Beim Laden eines Elements wird automatisch der entsprechende Parametersatz mitgeladen. Dadurch wird in komfortabler Weise eine spezifische Arbeitsumgebung zur Bearbeitung der selektierten Bibliothekshierarchie bzw. des selektierten Designobjekts aktiviert.

User Language

Im **Schaltplaneditor** ist der **Bartels User Language Interpreter** integriert, d.h. vom **Schaltplaneditor** aus können **User Language**-Programme gestartet werden. Der Anwender hat damit die Möglichkeit, eigene Zusatzfunktionen nach anwender- bzw. firmenspezifischen Bedürfnissen zu implementieren und in den **Schaltplaneditor** einzubinden. Hierzu zählen zum Beispiel Statusanzeigen und Parametereinstellungen, Report- und Testfunktionen (Fanoutkontrolle, Electronic Rule Check), Prüf- und Editierfunktionen, spezielle Plotfunktionen, Utilities zur Verwaltung von Bauteilbibliotheken, automatische Platzierungsfunktionen, firmenspezifische Batch-Prozeduren, usw. usf.

Im **Schaltplaneditor** können User Language Programme explizit oder implizit aufgerufen werden. Der explizite Programmaufruf erfolgt über den Menüpunkt Anwenderfunktion im Menü Datei. Nach der Aktivierung dieses Menüpunktes ist auf die Abfrage nach dem Programmnamen der Name des aufzurufenden **User Language**-Programms (z.B. **ulprog**) explizit einzugeben. Die Betätigung einer beliebigen Maustaste oder die Eingabe eines Fragezeichens **?** auf die Abfrage nach dem Programmnamen bewirkt hierbei die Aktivierung eines Popupmenüs mit allen aktuell verfügbaren **User Language**-Programmen.

User Language-Programme können auch implizit über die Tastatur aktiviert werden. Diese Art des Programmaufrufs ist immer dann möglich, wenn nicht gerade eine andere interaktive Eingabe über Tastatur erwartet wird. Die Spezifikation des Programmnamens erfolgt dabei implizit durch Drücken einer Taste. Zulässige Tasten sind dabei die Standardtasten (1, 2, ..., 0, a, b, c, ...; entsprechende Programmnamen sind **scm_1**, **scm_2**, ..., **scm_0**, **scm_a**, **scm_b**, **scm_c**, ...) bzw. die Funktionstasten (F1, F2, ...; entsprechende Programmnamen sind dabei **scm_f1**, **scm_f2**, ...).

Der **Schaltplaneditor** ermöglicht den ereignisgesteuerten Aufruf von **User Language**-Programmen. Dabei lösen spezielle Ereignisse bzw. Operationen implizit, d.h. automatisch den Aufruf von **User Language**-Programmen mit definierten Namen aus, sofern diese verfügbar sind. Im Einzelnen sind dies die **User Language**-Programme **SCM_ST** beim Starten des **Schaltplaneditor**, **SCM_LOAD** nach dem Laden eines Elements, **SCM_SAVE** vor dem Speichern eines Elements, **SCM_TOOL** bei Selektion eines Toolbarelements sowie **SCM_ZOOM** bei Änderung des Zoomfaktors. Der Aufruf über die Startupsequenz der Interpreterumgebung eignet sich besonders zur automatischen Voreinstellung von modulspezifischen Parametern sowie zur Tastaturprogrammierung und Menübelegung. Der implizite Aufruf von **User Language**-Programmen nach dem Laden bzw. vor dem Speichern von Elementen ermöglicht die automatische Aktivierung elementspezifischer Bearbeitungsparameter wie z.B. des zuletzt selektierten Zoombereichs oder spezieller Farbeinstellungen. Bei Interaktionen in der Werkzeugliste werden die den selektierten Toolbarelementen zugewiesenen Funktionen ausgelöst. Die Änderung des Zoomfaktors kann dazu benutzt werden, Aktualisierungen in Funktionen zur Verwaltung von Entwurfsansichten auszulösen.

Mit der **Bartels User Language** werden darüber hinaus mächtige Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Beachten Sie bitte, dass über die mit der BAE-Software ausgelieferten **User Language**-Programme eine Vielzahl von Zusatzfunktionen implementiert und transparent in die Benutzeroberfläche des **Schaltplaneditor**s eingebunden sind.

Eine ausführliche Beschreibung der **Bartels User Language** finden Sie im Bartels User Language Programmierhandbuch (Kapitel 4.2 enthält eine Auflistung aller mit der BAE-Software ausgelieferten **User Language**-Programme).

Neuronales Regelsystem

Im **Bartels AutoEngineer** sind eine Vielzahl mächtiger Zusatzfunktionen über das integrierte **Neuronale Regelsystem** implementiert. Kapitel 6.3.1 enthält eine Übersicht über die im Schalplanpaket bereitgestellten Regelsystemanwendungen.

2.2 Bibliotheksbearbeitung

Im Lieferumfang des **Bartels AutoEngineer** sind eine Reihe von Schaltzeichenbibliotheken enthalten. Natürlich kann jedoch der Fall eintreten, dass Sie für Ihren aktuell zu bearbeitenden Schaltplan ein Symbol benötigen, welches noch nicht in einer dieser mitgelieferten Bibliotheken enthalten ist. Nachfolgend wird anhand von Beispielen die Erstellung derartiger Bibliothekssymbole beschrieben. Dabei werden entsprechend der Datenbankhierarchie (siehe hierzu auch Kapitel 1.3) ausgehend von der untersten Hierarchieebene zunächst ein Pinsymbol (auf Markerebene), anschließend ein Stromlaufsymbol (auf Symbolebene) und schließlich noch zwei Labelsymbole (auf Labelebene) definiert. Alle diese Symbole werden in einem DDB-File mit Namen **`demo.ddb`** abgelegt. Gehen Sie hierzu zunächst in das bei der Software-Installation angelegte BAE-Jobs-Verzeichnis (z.B. **`c:\baejobs`**), und starten Sie den **AutoEngineer**:

```
>  c: ↵
>  cd c:\baejobs ↵
>  bae ↵
```

Rufen Sie das Schaltplan-Modul auf:



Sie befinden sich nun im Stromlauf-Editor des **Bartels AutoEngineer** und können mit der Erstellung der Bibliothekselemente beginnen. Bevor Sie jedoch eigene Stromlaufsymbole erstellen, sollten Sie sich mit den Normen zur Schaltzeichenerstellung vertraut machen. Häufig existieren firmenspezifische Konventionen, die zu beachten sind. Beispiele hierfür sind etwa das Raster für die Schaltzeichendarstellung, die Mindestbreite für das Schaltzeichen, die Kennzeichnung von Ein- und Ausgängen oder von Steuerblöcken, die Mindesthöhe für Texte, das Raster zur Platzierung der Pins, die Lage des Schaltzeichen-Nullpunktes, usw. Insbesondere der Nullpunkt und das Raster zur Platzierung der Pins sollte so grob (z.B. 2mm) gewählt werden, dass später in dem zur Schaltplanerstellung gewählten Raster (z.B. 1mm) die Verbindungen problemlos an diese Pins herangeführt werden können.

Abbildung 2-1 zeigt die SCM-Bibliothekssymbole, die wir in diesem Abschnitt erstellen werden.

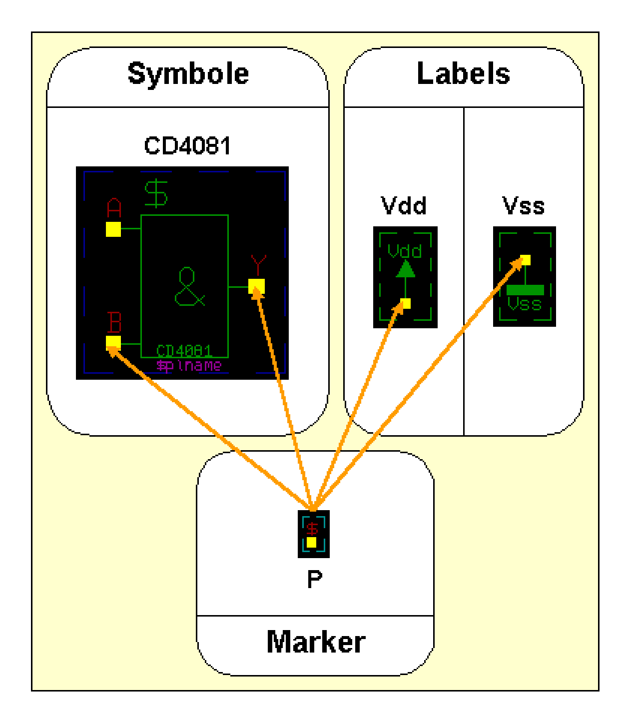
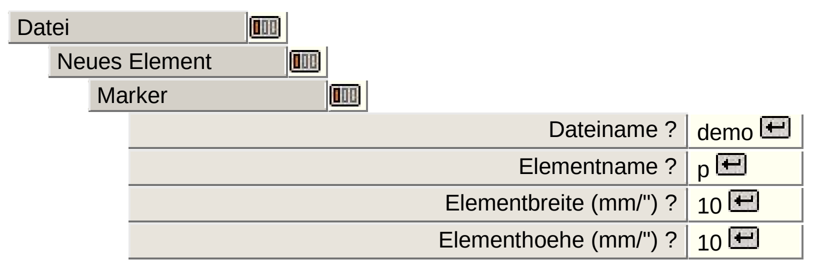


Abbildung 2-1: SCM-Bibliothekssymbole

2.2.1 Marker-Erstellung

Erzeugen des Markersymbols

Generieren Sie mit den folgenden Befehlen in der Datei **`demo.ddb`** einen Marker mit dem Elementnamen **`p`** und einer Elementgröße von 10*10 mm:

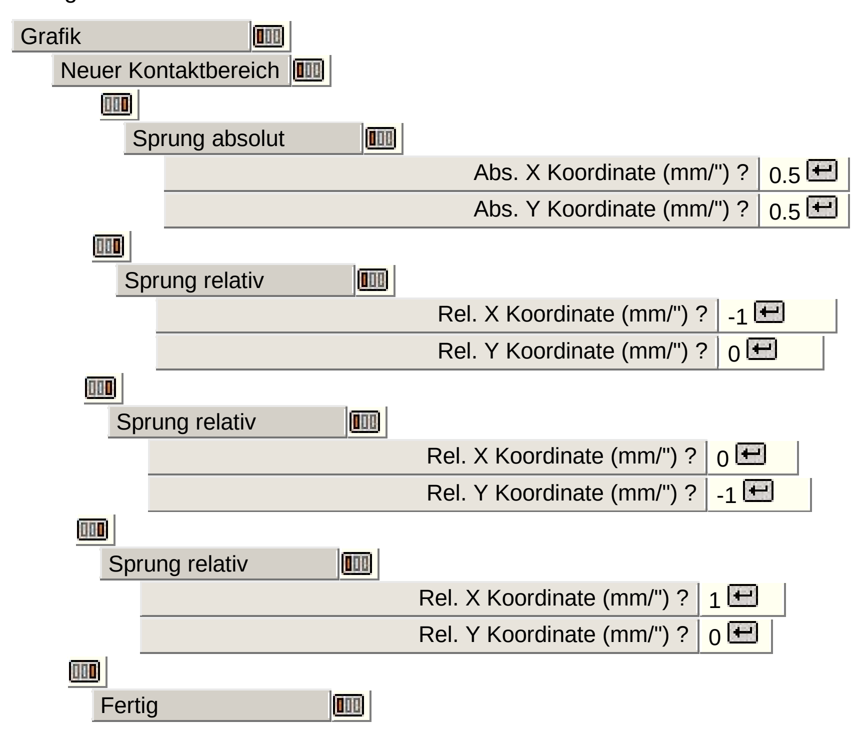


Auf dem Bildschirm sehen Sie nun einen quadratischen Rahmen mit einem Kreuz in der Mitte. Der Rahmen beschreibt die Elementgrenzen des Markers, während das Kreuz die Position des Element-Nullpunktes kennzeichnet.

Definieren des Kontaktbereiches

Pinsymbole dienen auf Planebene zur visuellen Überprüfung von Netzlistenänderungen, d.h. sobald eine Verbindung am Marker richtig angeschlossen wurde, wird der am Marker definierte Kontaktbereich nicht mehr auf dem Schema angezeigt. Die Kontaktflächen der Marker werden beim Plotten des Planes nicht mitgeplottet.

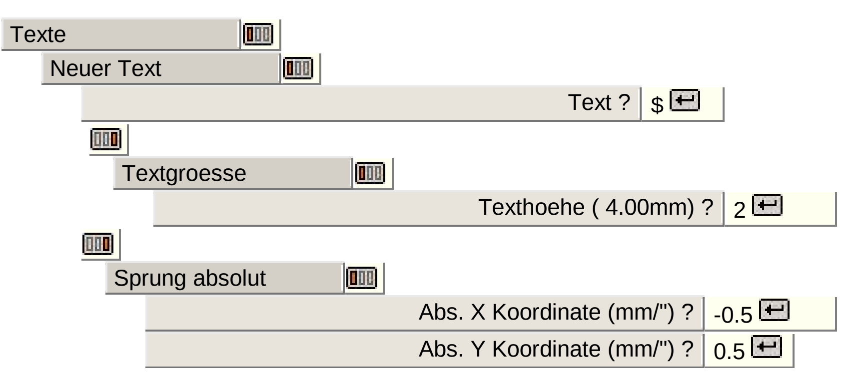
Definieren Sie nun im Nullpunkt des Markers einen quadratischen Kontaktbereich mit einer Kantenlänge von 1mm. Dies geschieht mit folgenden Befehlen:



Definieren der Referenz

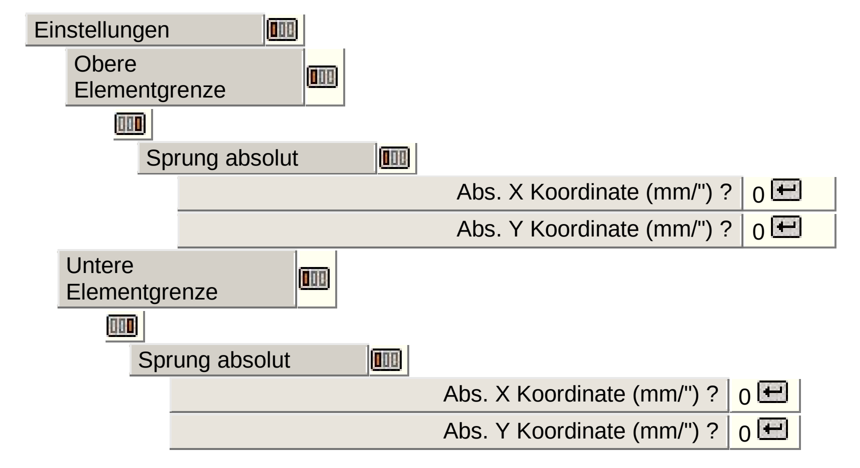
Wir haben nun bereits die grafische Gestaltung des Markersymbols abgeschlossen. Der Marker sollte jedoch noch mit einer Referenz für die Pinbezeichnung ausgestattet werden. Hierzu sei kurz die Bedeutung des **$**-Zeichens erklärt. Im **AutoEngineer** definiert das **$**-Zeichen eine Variable, die - auf einer Hierarchiestufe gesetzt - auf den höheren Hierarchieebenen den Namen des Elements anzeigt. So zeigt das auf Markerebene platzierte **$**-Zeichen auf Schaltzeichen- und Schaltplanebene den Pinnamen an, ein **$**-Zeichen auf Symbolebene platziert zeigt auf Schaltplanebene den Bauteilnamen an, usw.

Platzieren Sie nun mit folgenden Befehlen das **$**-Zeichen mit einer Texthöhe von 2mm an der Koordinate [-0.5,0.5]:



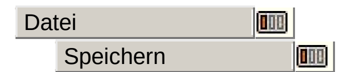
Definieren der Elementgrenzen

Nun sollten noch die Elementgrenzen des Markers so umdefiniert werden, dass sie die Objekte des Markers möglichst dicht umschließen. Dies geschieht wie folgt:

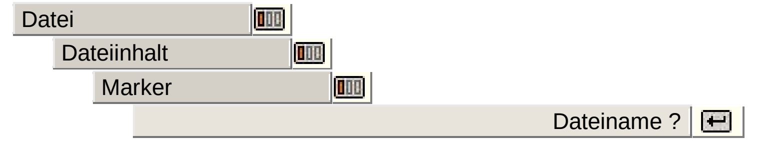


Sichern des erstellten Markersymbols

Speichern Sie das Markersymbol nun mit folgenden Befehlen ab:



Das Markersymbol ist nun definiert und unter dem Elementnamen **`p`** in der Datei **`demo.ddb`** abgespeichert. Sie können dies wie folgt überprüfen:



Da bei der Abfrage nach dem Dateinamen ein leerer String angegeben wurde, verwendet das System den Dateinamen des im Speicher befindlichen Elements (also **`demo.ddb`**). Das System sollte nun im Grafikarbeitsbereich folgende Liste mit den in der Datei **`demo.ddb`** enthaltenen Markersymbolen ausgeben:

```
Typ : Marker / Datei : demo.ddb

:p                   - Ende -
```

Betätigen Sie nun die Leertaste, um wieder in die Menüoberfläche zu gelangen.

2.2.2 Symbol-Erstellung

Wir wollen im Folgenden ein neues Bibliothekssymbol mit dem Elementnamen **CD4081** erstellen. Dieses Symbol kann in einer neuen oder in einer bereits bestehenden Bibliotheks- oder Projektdatei generiert werden. In diesem Beispiel erstellen wir das Symbol in unserer aktuellen Bibliotheksdatei **demo.ddb**.

Bei dem zu definierenden Bauteil **CD4081** bietet es sich an, nicht das komplette IC als ein Symbol zu erzeugen, sondern als Einzelgatter, da es innerhalb des Packages vier mal vorkommt. Wenn Sie die im Folgenden aufgeführten Arbeitsschritte richtig nachvollziehen, sollte sich das in Abbildung 2-2 dargestellte Symbol ergeben.

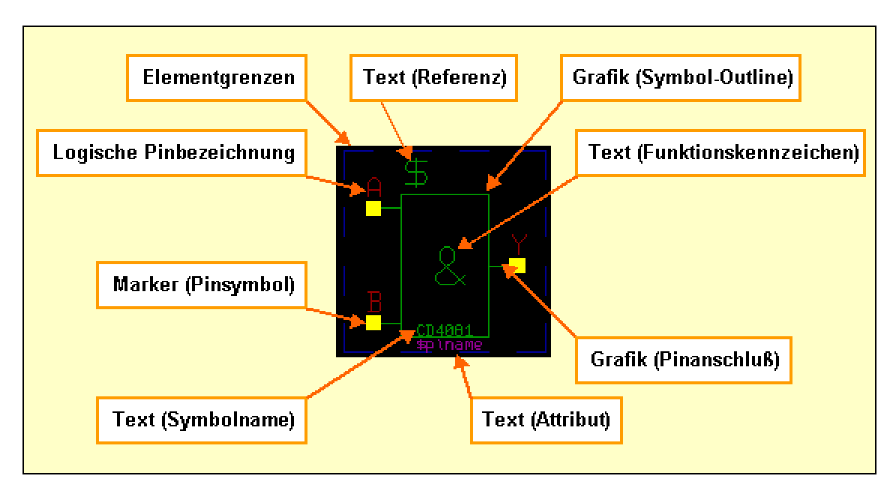
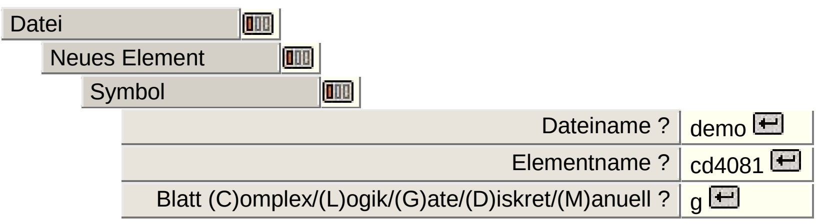


Abbildung 2-2: Stromlaufsymbol CD4081

Erzeugen des Stromlaufsymbols

Erzeugen Sie mit folgenden Befehlen ein neues Symbol (d.h. ein neues Element vom Typ Symbol) mit dem Elementnamen **CD4081** in der Datei **demo.ddb** (sofern vom vorhergehenden Arbeitsschritt noch ein Element der Datei **demo.ddb** geladen ist, genügt die Angabe eines leeren Strings (Betätigen der Eingabetaste ↵) auf die Abfrage nach dem Dateinamen):

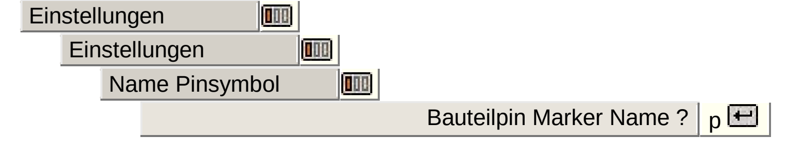


Die Symbolgröße legt vordefinierte Elementgrenzen für das Symbols fest. Die Elementgrenze ist der aktive Arbeitsbereich, innerhalb dessen Objekte platziert werden können. Dieser Bereich lässt sich auch nachträglich verändern. Sie können die Elementbreite und die Elementhöhe entweder manuell eingeben oder einfach eine der folgenden vordefinierten Elementgrößen auswählen:

Option	Element-/Symbolgröße
Complex	50 × 90 mm
Logik	30 × 30 mm
Gate	15 × 15 mm
Diskret	15 × 10 mm

Selektion des Pinsymbols

Selektieren Sie zunächst den zuvor erstellten Marker **p**, der für dieses Schaltzeichen als Pinsymbol verwendet werden soll:



Sie können mit Hilfe obiger Parameterzuweisung Schaltzeichen aus Pinsymbolen mit unterschiedlicher Geometrie erstellen. Per Default wird vom System der Marker mit dem Namen **pin** verwendet (achten Sie darauf, dass dieser Marker auch in der voreingestellten Bibliothek verfügbar ist; siehe hierzu auch die Beschreibung des Kommandos **SCMDEFLIBRARY** im Programm **BSETUP**).

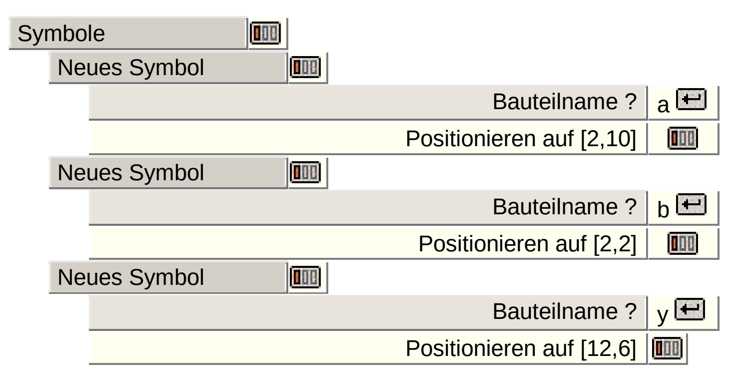
Platzieren der Pins

Nun sind die Pins des Symbols zu definieren. Das Bauteil **CD4081** beinhaltet 4 gleiche AND-Gatter mit den Eingängen **A** und **B** sowie dem Ausgang **Y**. Dies sind die logischen Namen der Pins (andere logische Pins könnten z.B. mit **INPUT**, **OUT**, **CLK**, etc. bezeichnet sein). Die Zuordnung zum jeweiligen physikalischen Pin erfolgt später mit Hilfe des Utilityprogramms **LOGLIB** (siehe hierzu auch die Beschreibung zum **Packager** in Kapitel 3.2 dieses Handbuchs). 1:1-Zuordnungen, d.h. die Identität logischer und physikalischer Pinbezeichnungen, sind natürlich ebenso möglich.

Die Marker des Schemasymbols sind die Bauteile, aus denen das Symbol interaktiv erstellt wird. Um in der Sprache des **AutoEngineer** zu bleiben: *Das Symbol des Symbols ist der Marker!*

Dieser zunächst etwas verwirrende Satz ist jedoch logisch. Ein Symbol wird aus Markern erstellt, welche die Pins des Symbols repräsentieren. Die Markerebene ist die unterste Hierarchieebene des Schematics. Folgender Zusammenhang ist gegeben: Der Marker ist das Element, durch dessen Platzierung auf der Symbolebene Symbole erzeugt werden, während durch das Platzieren der Symbole auf Planebene Schaltpläne erzeugt werden.

Platzieren Sie nun die Pins **A**, **B** und **Y** an den Koordinaten [2,10], [2,2] und [12,6] im Symbol:



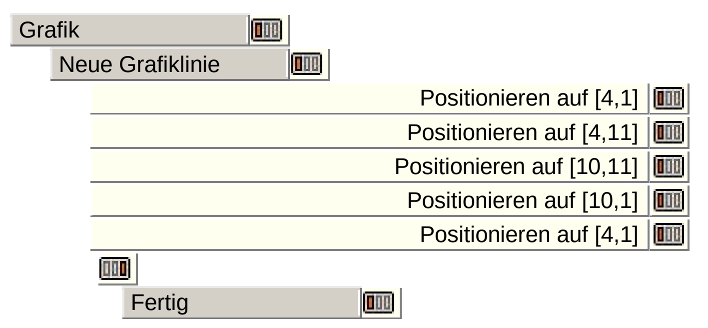
Die Marker sind nun platziert und stellen innerhalb des **CD4081**-Gatters die Pins **A**, **B** und **Y** dar. Die Koordinaten der Marker werden während der Platzierung im Statusfenster der Benutzeroberfläche angezeigt.

Die Eingabe eines **/** vor dem Bauteilnamen bewirkt die inverse Schreibweise des Namens. Damit ließe sich z.B. ein negierter Ausgang durch die Definition des Bauteilnamens **/y** kennzeichnen. Dies gilt auch für Texte.

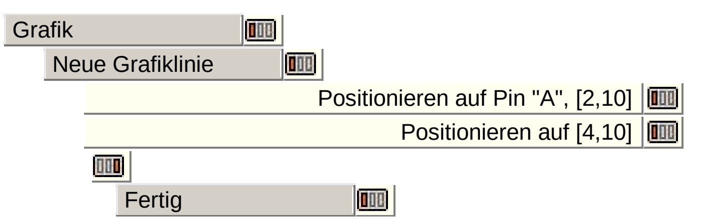
Grafik

Das Symbol besteht jetzt zwar aus allen für das Einzelgatter **CD4081** benötigten Pins, aber noch nicht aus der Symbolgeometrie, wie Sie entsprechend Ihren Bedürfnissen oder nach Norm in der Schaltung erscheinen soll.

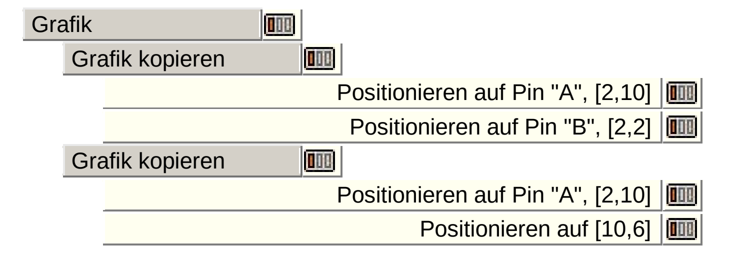
Erstellen Sie jetzt mit folgenden Kommandos die Symbol-Outline des **CD4081**:



Um das Symbol zu komplettieren, sind jetzt die Pins als Grafiklinien oder Flächen einzuzeichnen. Gehen Sie wie folgt vor:



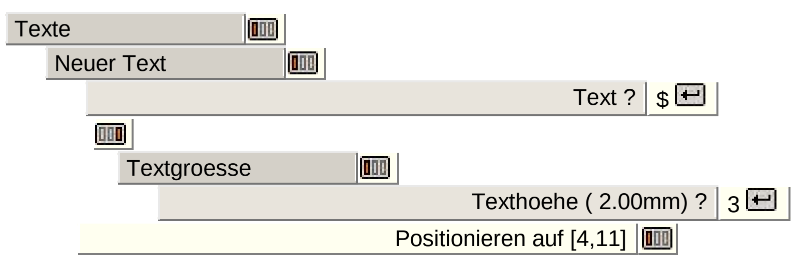
Nun ist eine waagrechte Grafiklinie ausgehend vom Mittelpunkt des Pins **A** bis zur Symbol-Outline eingezeichnet. Diese Linie ist auch nach Anschluss des Markers **A** im Plan als Pin sichtbar und wird mitgeplottet. Kopieren sie nun die am Pin **A** gezeichnete Grafiklinie auf die Pins **B** und **Y**:



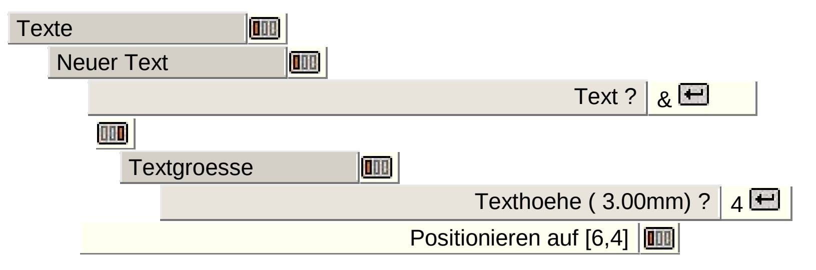
Text und Attribute

Die grafische Gestaltung des neuen Symbols ist nun abgeschlossen. Das Symbol sollte jedoch noch mit einigen Texten ausgestattet werden. Zunächst setzen wir die Variable für die Symbol-Referenz, d.h. wir legen fest, an welcher Stelle am Symbol der Bauteilname (IC01, R20, V2) erscheinen soll. Hierzu sei nochmals kurz die Bedeutung des **$**-Zeichens erklärt. Im **AutoEngineer** wird durch den Text **$** eine Variable definiert, die auf den nächst höheren Hierarchieebenen den Namen des Elements anzeigt. Ein **$** als Text auf Symbolebene platziert zeigt also auf Planebene den Bauteilnamen an.

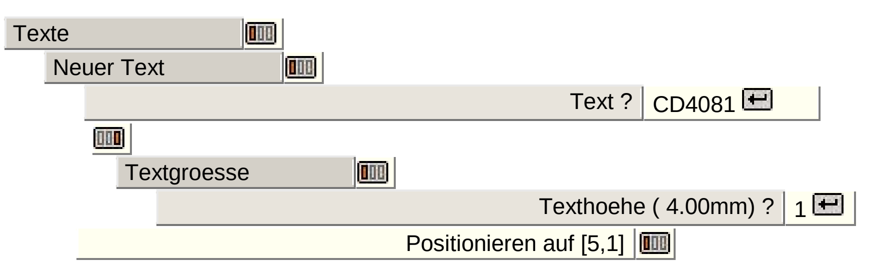
Platzieren Sie nun den Text **$** mit einer Textgröße von 3mm:



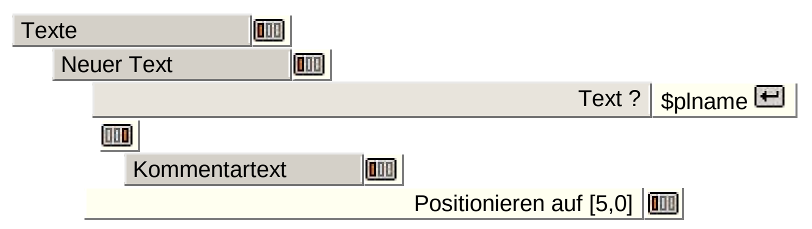
Zur Kenntlichmachung der Funktion des Symbols im Plan sollte noch ein **&**-Zeichen für das AND-Gatter im Symbol platziert sein.



Als nächstes tragen Sie mit folgender Befehlsfolge den Gatternamen **CD4081** (Texthöhe 1mm) in das Symbol ein:



Definieren Sie nun das Attribut **$plname** als Kommentartext (Texthöhe 1mm ist noch eingestellt):



Im Schaltplanpaket des **Bartels AutoEngineer** wird unterschieden zwischen Standardtext und Kommentartext. Sowohl Standardtext als auch Kommentartext werden auf dem Bildschirm dargestellt, bei der Generierung von Plots werden jedoch nur die Standardtexte ausgegeben. Somit besteht über die Definition von Kommentartexten die Möglichkeit, Attribute oder Texte beim Plotten auszublenden.

Es besteht auch die Möglichkeit, Texte mit Rahmen zu versehen. Die Einstellung erfolgt beim Platzieren des Textes in den über die rechte Maustaste erreichbaren Untermenüs der Funktionen Neuer Text, Text bewegen bzw. Text kopieren über die Optionen Rahmen 1, Rahmen 2 und Offene Rahmen. Rahmen 1 erzeugt einen Textrahmen im Abstand von 1/8 der Texthöhe, Rahmen 2 erzeugt einen Textrahmen im Abstand von 1/4 der Texthöhe. Bei Aktivierung offener Rahmen wird die am Textursprung befindliche vertikale Rahmenlinie offen gelassen; dadurch können Textfahnen für Labels erzeugt werden, die sich dynamisch der Textlänge anpassen. Die Optionen Rahmen 1 und Rahmen 2 können simultan angewendet werden. Bei der Plotausgabe kann es dazu kommen, dass Textnegierungsstriche mit Textrahmen verschmelzen. Die Gefahr hierzu besteht insbesondere beim Plotten kleiner Texte mit großen Stiften. Mit der Option Standardtext werden alle Textrahmeneinstellungen zurückgesetzt.

Die über die rechte Maustaste in den Funktionen Neuer Text, Text bewegen und Text kopieren erreichbaren Untermenüs enthalten auch die Optionen Keine Rotation und Zentriert. Damit Keine Rotation wird die Möglichkeit der Drehung für den bearbeiteten Text abgeschaltet. Texte, für die der Modus Keine Rotation gesetzt ist, werden in allen Datenbankhierarchieebenen immer ungedreht angezeigt bzw. geplottet. Mit Zentriert wird der Text zentriert an der Platzierungsposition ausgerichtet. Um diese Modi wieder abzuschalten, ist die Option Standardtext aus demselben Untermenü auf den Text anzuwenden.

Attribute werden im **AutoEngineer** als Texte bestehend aus dem **$**-Zeichen und nachfolgend dem (kleingeschriebenen) Attributnamen auf Symbolebene definiert. Diese Attribute haben die Funktion von Variablen, d.h. derartigen Attributen können auf Planebene Werte zugewiesen werden. Die in Tabelle 2-1 aufgeführten Attribute besitzen aufgrund programmbedingter Konventionen eine besondere Bedeutung.

Tabelle 2-1: Spezielle Attribute im Bartels AutoEngineer

Attributname	Funktion/Bedeutung
$	Referenzname
$$	Logischer Referenzname
$llname	Logical Library Name
$plname	Physical Library Name (und alternative Layoutgehäusebauformen)
$ulname	Used Library Name (alternative Layoutgehäusezuweisung)
$rpname	Requested Part Name
$rbname	Requested Backannotated Part Name
$gp	Gate Pin
$blkname	Hierarchieblock-Name
$pageref	SCM-Label-Stromlaufblatt-Namensliste
$pagecref	SCM-Label-Stromlaufblatt-Kommentarliste
$orgname	Ursprüngliche(r)/interne(r) SCM-Symbol-/Bauteilname(n) eines Layoutbauteils
$pagename	Schaltplanbattname(n) eines Layoutbauteils
$blkrname	Schaltplanblockname(n) eines Layoutbauteils
$rlname	Requested Logical Library Name
$rlext	Requested Logical Library Name Extension
$val	Wert
$pow	Leistung
$type	Bauteiltyp
$comment	Kommentar (englisch)
$commentge	Kommentar (deutsch)
$manufacturer	Hersteller
$partside	Bauteilseite (**top** = ungespiegelt, **bottom** = gespiegelt)
$pltbaeversion	**Bartels AutoEngineer** Software-Versionsnummer (nur Lesezugriff)
$pltbaebuild	**Bartels AutoEngineer** Software-Buildnummer (nur Lesezugriff)
$pltfname	Projektdateipfadname

Attributname	Funktion/Bedeutung
`$Pltfname`	Projektdateipfadname (Großschreibung)
`$pltfsname`	Projektdateiname (ohne Verzeichnispfad)
`$Pltfsname`	Projektdateiname (Großschreibung, ohne Verzeichnispfad)
`$pltpagecnt`	Gesamtanzahl Projektschaltplanblätter
`$pltename`	Elementname
`$Pltename`	Elementname (Großschreibung)
`$pltdatede`	Aktuelles Datum (deutsches Format)
`$pltdateus`	Aktuelles Datum (US-amerikanisches Format)
`$pltdate2de`	Aktuelles Datum (deutsches Format; zweistellige Jahreszahl)
`$pltdate2us`	Aktuelles Datum (US-amerikanisches Format; zweistellige Jahreszahl)
`$plttime`	Aktuelle Uhrzeit
`$pltsdatede`	SCM-/Layout-Sicherungsdatum (deutsches Format)
`$pltsdateus`	SCM-/Layout-Sicherungsdatum (US-amerikanisches Format)
`$pltsdate2de`	SCM-/Layout-Sicherungsdatum (deutsches Format; zweistellige Jahreszahl)
`$pltsdate2us`	SCM-/Layout-Sicherungsdatum (US-amerikanisches Format; zweistellige Jahreszahl)
`$pltstime`	SCM-/Layout-Sicherungsuhrzeit
`$pltpname`	Layoutelementname des letzten **Packager**-Laufs
`$Pltpname`	Layoutelementname des letzten **Packager**-Laufs (Großschreibung)
`$pltpdatede`	Datum (deutsches Format) des letzten **Packager**-Laufs
`$pltpdateus`	Datum (US-amerikanisches Format) des letzten **Packager**-Laufs
`$pltpdate2de`	Datum (deutsches Format; zweistellige Jahreszahl) des letzten **Packager**-Laufs
`$pltpdate2us`	Datum (US-amerikanisches Format; zweistellige Jahreszahl) des letzten **Packager**-Laufs
`$pltptime`	Uhrzeit des letzten **Packager**-Laufs
`$pltcname`	Layoutelementname
`$pltcdatede`	Datum des letzten Namensupdates (deutsches Format)
`$pltcdateus`	Datum des letzten Namensupdates (US-amerikanisches Format)
`$pltcdate2de`	Datum des letzten Namensupdates (deutsches Format; zweistellige Jahreszahl)
`$pltcdate2us`	Datum des letzten Namensupdates (US-amerikanisches Format; zweistellige Jahreszahl)
`$pltctime`	Uhrzeit des letzten Namensupdates
`$pltfbname`	Elementspezifischer Projektdateiname ohne die Endung `.ddb`
`$Pltfbname`	Elementspezifischer Projektdateiname ohne die Endung `.ddb` (Großschreibung)
`$pltfbsname`	Elementspezifischer Projektdateiname ohne die Endung `.ddb` und ohne den Verzeichnispfad
`$Pltfbsname`	Elementspezifischer Projektdateiname ohne die Endung `.ddb` und ohne den Verzeichnispfad (Großschreibung)
`$pltecomment`	DDB-Elementkommentar - Elementspezifischer Kommentartext
`$drcblk`	Design Rule Check Block (Pin/-Netzattribut)
`$net`	Pinnetzname
`$netname`	Netzname (Netzattribut)
`$nettype`	Netztyp (Pin-/Netzattribut)
`$powpin`	Versorgungspindefinition (Pinattribut)
`$viastk`	Viapadstacktyp (Netzattribut)

Attributname	Funktion/Bedeutung
$notest	Deaktivierung automatische Testpunktgenerierung (Netzattribut)
$routdis	Deaktivierung Autorouting (Netzattribut)
$layers	Autoroutinglagen (Netzattribut)
$@	Layoutbauteilname (Padstackattribute)

Die Systemattribute **$pltbaeversion** und **$pltbaebuild** dienen der Anzeige bzw. Ausgabe der Versions- und Buildnummer der **Bartels AutoEngineer**-Software. Diese ermöglicht die für ISO-Zertifizierungen benötigte Dokumentation des verwendeten Softwarestandes auf Schaltplanausgaben.

Die Attribute **$**, **$llname**, **$plname**, **$gp**, **$ulname** und **$blkname** werden vom System automatisch gesetzt. Die Referenz (**$**) wird auf der jeweils nächsthöheren Hierarchieebene durch den Namen des Elements ersetzt; auf Schaltplanebene wird hierfür der durch den **Packager** erzeugte Bauteil- bzw. Pinname eingesetzt. Auf Symbolebene kann über das Attribut **$$** der logische Bauteilname zur Anzeige auf Schaltplanebene auch nach dem **Packager**-Lauf referenziert werden, und auf Markerebene kann über das Attribut **$$** entsprechende die logische Pinbezeichnung zur Anzeige auf Schaltplanebene auch nach dem **Packager**-Lauf referenziert werden.

Auf Symbolebene platzierte **$llname**-Texte werden auf Schaltplanebene durch den Namen des Symbolmakros ersetzt. Das **$llname**-Attribut kann nicht interaktiv bzw. explizit gesetzt werden, da der **Packager** automatisch die korrekte Symbolnamenszuweisung vornimmt. Die Dokumentation des Symbolnamens mit Hilfe von **$llname**-Texten erleichtert die Erstellung neuer Schaltplansymbole aus existierenden Symbolen, da der **$llname**-Text des neuen Symbols auf Schaltplanebene automatisch durch dessen neuen Namen ersetzt wird und ein Editieren des Textes daher unnötig ist.

Im Layout kann **$llname** zur Visualisierung des Schaltzeichen-Namens verwendet werden. Außerdem überträgt der **Packager** die Namen der SCM-Symbolpins automatisch auf das Pinattribut **$llname** welches im Layout durch die Definition entsprechender Texte auf Padstackebene visualisiert werden kann.

Über den Physical Library Name (**$plname**) kann auf Schaltplanebene ein Symbol abweichend von der vordefinierten Defaultzuweisung aus der Logischen Bibliothek (siehe hierzu auch die Beschreibungen des Utilityprogramms **LOGLIB** sowie des **Packagers**). in ein anderes Gehäuse zugewiesen werden. So kann z.B. durch Zuweisung des Wertes **so14** an das Attribut **$plname** die Verwendung der SMD-Gehäusebauform **so14** (anstelle der bedrahteten Bauform) **dil14**) erzwungen werden. Durch eine **$plname**-Attributwertzuweisung der Form **[dil14,so14]** kann wahlweise auch eine Auswahlliste möglicher Alternativbauformen an das Layout übergeben werden.

Die Backannotation überträgt die im Layout zugewiesene Gehäusebauform in das **$ulname**-Attribut (Used Library Name), D.h., über dieses Attribut können im Stromlauf alternative Layoutgehäusezuweisungen angezeigt bzw. abgefragt werden.

Mit dem Requested Part Name (**$rpname**) können Symbole auf Schaltplanebene bestimmten Bauteilen zugewiesen werden (z.B. **IC4** anstelle der durch den **Packager** automatisch generierten Zuweisung); diese Funktion wird benötigt, um die Zuweisung bestimmter Schaltplansymbole an definierte Gehäuse zu erzwingen (z.B. bei Relais oder Mehrfach-Operationsverstärkern).

Mit dem Requested Backannotated Part Name (**$rbname**) können wie mit **$rpname** Symbole auf Schaltplanebene bestimmten Bauteilen zugewiesen werden. Mit **$rbname** zugewiesene Layoutbauteilnamen können aber im Gegensatz zu **$rpname**-Zuweisungen im **Layouteditor** geändert werden und Backannotation überträgt geänderte Layoutbauteilnamen zurück in die **$rbname**-Attribute der zugehörigen Schaltplansymbole.

Mit Hilfe des Attributs **$gp** (Gate Pin) ist es möglich, bei Mehrfachsymbolen im Schaltplan explizit vorzugeben, welche Position ein Gatter im Layoutbauteil einnimmt. Das Attribut **$gp** ist dazu auf den Namen des ersten im **xlat**-Kommando der logischen Definition aufgeführten Layoutbauteilpins des gewünschten Gatters zu setzen (siehe hierzu auch **LOGLIB**). Um Überbelegungskonflikte zu vermeiden, sollte das **$gp**-Attribut immer entweder für alle Gatter oder kein Gatter eines Layoutbauteiles gesetzt werden. Das **$gp**-Attribut ermöglicht z.B. bei Mehrfach-Operationsverstärkern eine spezifische Zuordnung der einzelnen Komponenten. Darüberhinaus können durch Zuweisungen von Pinbezeichnungen an dieses Attribut Steckerpinsymbole für die Anschlüsse mehrpoliger Stecker definiert werden, sofern die Steckeranschlüsse über **xlat** jeweils als Einzelpin-Gatter definiert sind. Mit dem **$gp**-Attribut zugewiesene Gatter und Pins sind vom Pin- und Gattertausch ausgeschlossen.

Der Hierarchie-Blockname (**$blkname**) wird vom **Packager** bei der Bearbeitung hierarchisch aufgebauter Schaltpläne automatisch mit dem Namen des Hierarchie-Blocks besetzt, in dem das entsprechende Bauteil definiert wurde.

Über das Attribut **$pltpagecnt** kann im Schaltplan die Gesamtanzahl der im aktuellen Projekt enthaltenen Schaltplanblätter angezeigt werden.

Auf Labelebene platzierte **$pageref**-Texte werden auf Schaltplanebene durch die Namensliste der Schaltplanblätter ersetzt, auf denen das durch den Label angegebene Netz verwendet wird.

Auf Labelebene platzierte **$pagecref**-Texte werden auf Schaltplanebene durch die Kommentare der Schaltplanblätter ersetzt, auf denen das durch den Label angegebene Netz verwendet wird.

Die Attribute **$orgname** (ursprüngliche(r)/interne(r) SCM-Symbol-/Bauteilname(n) bzw. SCM-(Teil-)Netzname), **$pagename** (Schaltplanbattname(n)) und **$blkrname** (Schaltplanblockname(n)) zur Bestimmung der Herkunft aus hierarchischen Schaltplanblöcken werden vom **Packager** automatisch an Layoutbauteile zugewiesen.

Über eine Bauteildefinitionszuweisung an das Symbolattribut **$rlname** (Requested Logical Library Definition) kann eine vom Symbolnamen abweichende Bauteildefinition aus der Logischen Bibliothek referenziert werden. Dies ermöglicht z.B. die Zuweisung spezifischer Gehäusebauformen mit verschiedenen Festattributen für Sachnummern an Schaltplansymbole. Um Fehlzuweisungen zu vermeiden, müssen die referenzierten Bauteildefinitionen mit Hilfe des entsprechenden **LOGLIB**-Eintrags der gleichen Bauteilklasse zugeordnet sein (siehe hierzu auch Kapitel 7.11). Bei **mainpart/subpart**-Symbolen ist zu beachten, dass diese nur korrespondierend gewechselt werden können. D.h., bei Definitionen wie z.B. **amain/asub** und **bmain/bsub** dürfen die Symbole, die in ein Gehäuse gepackt werden, entweder **amain/asub** oder **bmain/bsub** referenzieren, die Kombinationen **amain/bsub** und **bmain/bsub** sind hingegen nicht zulässig.

Mit **$rlext** (Requested Logical Library Name Extension) können von Symbolnamen abweichende Bauteildefinitionszuweisungen ähnlich wie mit **$rlname** vorgenommen werden. **$rlext**-Einträge weisen den **Packager** an, logische Bibliotheksdefinitionen mit spezifischen Namenserweiterungen (vom Basisnamen getrennt durch Unterstrich _) an die entsprechenden Symbole zuzuweisen. Dies ermöglicht z.B. die Zuweisung spezifischer standardisierter bzw. herstellerspezifischer Gehäusebauformen mit verschiedenen Festattributen für Sachnummern an Schaltplansymbole.

Die optionalen Attribute **$val** (Value/Wert), **$pow** (Power/Leistung), **$type** (Bauteiltyp), **$comment** (Kommentar englisch), **$commentge** (Kommentar deutsch) und **$manufacturer** (Hersteller) sind in den mitgelieferten Bibliotheken des **Bartels AutoEngineer** zum Teil vordefiniert. Die Definition dieser Attribute bzw. die Eintragung der entsprechenden Attributwerte können nach Bedarf vom Anwender durchgeführt werden.

Über das Bauteilattribut **$partside** kann für ein Bauteil vorgegeben werden, ob es nur ungespiegelt (Attributwert **top**) oder nur gespiegelt (Attributwert **bottom**) platziert werden darf. Beim manuellen und automatischen Platzieren hat diese Vorgabe Vorrang vor anderen Spiegelungseinstellungen. Entgegen der Vorgabe platzierte Bauteile werden vom DRC markiert und im Utilities / Report als Bauteilseitenfehler gelistet. Die Bibliothek ROUTE enthält das Tagsymbol tag_sym_partside zum Setzen des Attributs **$partside**.

Die Attribute **$plttime** (aktuelle Uhrzeit), **$pltdatede** (aktuelles Datum, deutsche Notation) und **$pltdateus** (aktuelles Datum, US-Notation) werden bei geladenem Schaltplan bzw. Layout bei der Bilddarstellung und bei der Plotausgabe jeweils durch die aktuelle Uhrzeit bzw. das aktuelle Datum ersetzt. Die Attribute **$pltstime** (Layout-Sicherungsuhrzeit), **$pltsdatede** (Layout-Sicherungsdatum, deutsche Notation) und **$pltsdateus** (Layout-Sicherungsdatum, US-Notation) werden bei der Bilddarstellung und bei der Plotausgabe jeweils durch die Uhrzeit bzw. das Datum der zuletzt für das aktuell geladene Layout durchgeführten Sicherung ersetzt. Bei der Anzeige der Uhrzeit- und Datumsangaben spielt es keine Rolle, auf welcher Datenbankebene (Marker, Symbol/Label, Plan bzw. Pad, Padstack, Bauteil, Layout) der jeweilige Attributtext definiert ist. Existiert ein gesetztes herkömmliches Attribut mit gleichem Namen auf einer untergeordneten Hierarchieebene (Symbol/Marker bzw. Bauteil/Padstack) bzw. Bauteil so besitzt dieses Priorität bei der Anzeige bzw. Ausgabe.

Die Attribute **$pltpname** (Layoutelementname), **$pltpdatede** (Datum, deutsche Notation), **$pltpdateus** (Datum, US-Notation) und **$pltptime** (Uhrzeit) werden bei geladenem Schaltplan bzw. Layout bei der Bilddarstellung und bei der Plotausgabe jeweils durch den Layoutelementnamen bzw. die Uhrzeit bzw. das Datum des zuletzt durchgeführten **Packager**-Laufs ersetzt.

Die Attribute **$pltdate2de**, **$pltdate2us**, **$pltsdate2de**, **$pltsdate2us**, **$pltpdate2de** und **$pltpdate2us** sind Varianten mit zweistelliger Jahreszahlanzeige der Attribute **$pltdatede**, **$pltdateus**, **$pltsdatede**, **$pltsdateus**, **$pltpdatede** und **$pltpdateus**.

Die Attribute **$pltcname**, **$pltcdatede**, **$pltcdate2de**, **$pltcdateus**, **$pltcdate2us** und **$pltctime** dienen der Anzeige des Layoutelementnamens sowie des Datums und der Uhrzeit des zuletzt durchgeführten Namensupdates. Im Gegensatz zu den Attributen zur Anzeige von **Packager**-Daten werden diese Einträge zusätzlich auch bei der Durchführung einer Backannotation geändert.

Mit den Attributtexten **$pltfbname** (File Base Name) bzw. **$pltfbsname** (File Base Short Name) ist es möglich, den Namen der Projektdatei auf dem aktuell geladenen Element anzuzeigen. **$pltfbname** zeigt den Projektdateinamen ohne die Endung **.ddb** an. **$pltfbsname** zeigt den Projektdateinamen ohne die Endung **.ddb** und ohne den Verzeichnispfad an.

Das Systemattribut **$pltecomment** (Elementkommentar) dient der Zuweisung von Kommentaren an DDB-Elemente. Die Zuweisung erfolgt wahlweise über das Untermenü Datei / Elementkommentar oder für das aktuell geladene Element auch in der Dialogbox Einstellungen / Einstellungen. Bei der Auswahl von DDB-Dateielementen wird dieser Kommentar neben dem Elementnamen angezeigt. Bei der PDF-Ausgabe von Schaltplänen wird bei vorhandenem Schaltplanblattkommentar der Kommentar anstatt des Blattnamens in das Seiteninhaltsverzeichnis des PDF-Dokumentes übernommen.

Das Netzattribut **$netname** wird vom **Packager** als Netzname an das angeschlossenen Netz übertragen und hat Vorrang vor eventuell per Label zugewiesenen Netznamen. Damit ist eine eindeutige Benennung auch für solche Netze mit unterschiedlich benannten Labels möglich.

Das Pinattribut **$nettype** wird vom **Packager** automatisch von Pins auf die angeschlossenen Netze übertragen. Sind an einem Netz Pins mit unterschiedlichen Attributwerteinträgen für **$nettype** angeschlossen so wird für das Netz der Wert **mixed** eingetragen. Das Pinattribut **$drcblk** wird vom **Packager** automatisch von Pins auf die angeschlossenen Netze übertragen. Der Attributwerteintrag für **$drcblk** addressiert in **BAE HighEnd** einen Parameterblock mit Entwurfsregeln, der damit an das entsprechende Netz zugewiesen wird.

Das Pinattribut **$powpin** steuert, dass ein Symbolpin ebenso wie über **net**-Kommandos definierte Pins seine Anschlussbreite aus dem Netzattribut für die Versorgungspinanschlussbreite bezieht. Das Tagsymbol tag_pin_powerpin aus der Bibliothek ROUTE kann zum Setzen dieses Pinattributs verwendet werden. Alternativ kann das Pinattribut **$powpin** in der Symbollogik auch fest zugewiesen werden:

```
newattr "$powpin" = "1" : (pinname);
```

Das Netzattribut **$viastk** dient der Zuweisung von Viatypen an Netze. Netzspezifische Viapadstackzuweisungen werden vom **Autorouter** entsprechend berücksichtigt. Durch die Zuweisung des Netzattributs **$notest** können spezifische Netze von der automatischen Testpunktgenerierung durch den **Packager** ausgenommen werden. Durch die Zuweisung des Netzattributs **$routdis** können spezifische Netze vom Autoroutingprozess ausgenommen werden. Über das Netzattribut **$layers** können netzspezifische Routinglagen (durch Komma getrennte Signallagennummern) für den Autoroutingprozess in **BAE HighEnd** an spezifische Netze zugewiesen werden.

Das Attribut **$@** ist padstack- bzw. layoutspezifisch. Ein auf Padstackebene platzierter **$@**-Text wird auf Layoutebene durch den Namen des Bauteils ersetzt, auf dem der Padstack als Pin platziert ist. Damit ist es möglich, für außerhalb eines Bauteils platzierte Pins den Namen des zugehörigen Bauteils am Pin zu dokumentieren (z.B. für die Funktion Pin bewegen).

Ein auf Padstackebene platzierter **$@**-Text wird auf Layoutebene durch den Namen des Bauteils ersetzt, auf dem der Padstack als Pins platziert ist. Damit ist es z.B. möglich bei mit Pin bewegen ausserhalb eines Bauteils platzierten Pins den Namen des zugehörigen Bauteils am Pin zu dokumentieren.

Neben den in Tabelle 2-1 aufgeführten Attributen können beliebige andere Attribute definiert werden (typische Beispiele hierfür sind **$tolerance**, **$identno**, **$sachnummer**, **$preis**, **$lieferant**, **$delay**, **$bauteilhoehe**, usw.).

Mit der Funktion Attribut bewegen aus dem Menü Texte bzw. Symbole können Symbolattribute bewegt bzw. platziert werden. Die mit Attribut bewegen festgelegten Textoffsets haben Vorrang vor ggf. mit Name bewegen vorgegebenen, globalen Textoffsets. Die Selektion des zu verschiebenden Attributs erfolgt durch Anklicken des Attributtexts. Der Symbolname selbst gilt ebenfalls als Attribut und kann somit mit der Funktion Attribut bewegen auch selektiv verschoben werden, ohne die Platzierung der übrigen Symbolattribute zu beeinträchtigen.

Attributwerte können im **Schaltplaneditor** mit der Funktion Wert zuweisen aus dem Menü Symbole oder durch entsprechende Einträge in der Bauteilbibliothek an Bauteile zugewiesen werden.

In den Windows- und Motifversionen aktiviert die Funktion Wert zuweisen eine Dialogbox für die Attributwerteingabe. Es werden bis zu 12 Attribute simultan dargestellt. Bei mehr als 12 Attributen kann über die Schaltflächen Weitere und Vorherige zwischen mehreren Attributseiten hin- und hergeblättert werden. In der Dialogbox ist jedem Attribut eine Zeile zugeordnet. Diese besteht aus der Schaltfläche Kein Wert, dem Attributwerteingabefeld und dem Attributnamenslabel. Mit Kein Wert wird der Attributwert komplett zurückgesetzt. Die Rücksetzung wird durch Anzeige von **!nicht_gesetzt!** für den Attributwert signalisiert. Die Rücksetzung unterscheidet sich von der Eingabe eines Leerstrings für den Attributwert. Ein Leerstring wird als solcher beim Attribut eingetragen und erscheint auch in der Netzliste. Nicht gesetzte Attribute werden überhaupt nicht in die Netzliste übertragen.

Die zugewiesenen Attribute werden in die Netzliste eingetragen und durch den **Packager** in das Layousystem übertragen. Im Layout können Attribute durch entsprechende Textdefinitionen (z.B. auf Dokumentarlagen) visualisiert werden. Auch können die entsprechenden Werteinträge mit **User Language**-Programmen oder mit Hilfe des Utilityprogramms **USERLIST** praktisch beliebig weiter in Richtung Fremdsysteme, CAM, PPS, usw. ausgewertet werden.

Symbolnamensmuster

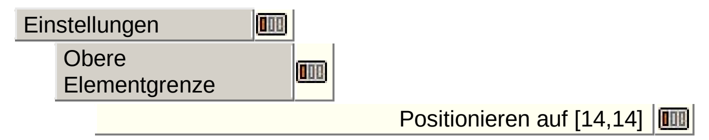
Als nächstes sollte dem System das gewünschte Muster für die automatische Bauteilbenennung im Schaltplan mitgeteilt werden:



Hiermit ist das Muster für die automatische Bauteilbenennung auf **IC** und 2 Stellen für die Nummerierung festgelegt. Wird kein Muster definiert, so gilt der Defaultwert **N????**, d.h. es werden Bauteilnamen beginnend mit dem Buchstaben **N** und nachfolgend 4 Stellen für die Nummer vergeben. Die Startnummer für die automatische Bauteilnummerierung ergibt sich zu 10 beim Symbolnamensmuster **??**, 100 bei **???**, 1000 bei **????**, usw. Damit ist i.d.R. sichergestellt, dass sich bei einer später im Layout notwendigen Bauteilumbenennung keine Namenskonflikte ergeben können. Eine solche Bauteilumbenennung wird z.B. durchgeführt, um die Lesbarkeit des Bestückungsplans (insbesondere für die Handbestückung) zu verbessern. Dies ist üblicherweise wichtiger, als eine nach irgendwelchen erdachten Regeln durchgeführte Benennung der Stromlaufsymbole auf dem Schaltplan (die Funktion des Bauteils bzw. der Schaltung ist hier ja bereits am Symbol abzulesen). Daher empfehlen wir, bei der Platzierung der Stromlaufsymbole die Funktion zur automatischen Bauteilbenennung zu benutzen und erst im Layout die endgültige Nummerierung zu erzeugen. Entsprechende Funktionen - auch zur automatischen Bauteilbenennung - stehen sowohl im **Layouteditor** als auch im **Autoplacement** zur Verfügung.

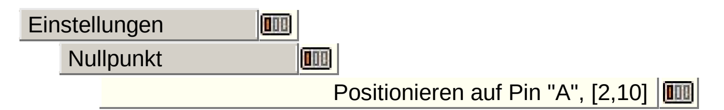
Definieren der Elementgrenzen

Setzen Sie nun die rechte obere Ecke der Elementgrenze auf die Koordinate [14,14]:



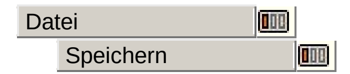
Definieren des Nullpunkts

Der Nullpunkt des Symbols ist auf Planebene dessen Pickpunkt. Das ist der Punkt des Symbols, der am Fadenkreuz hängt. Am zweckmäßigsten ist es, diesen Punkt auf einen der Pins (in unserem Fall z.B. Pin **A**, Koordinate [2,10]) zu legen:



Sichern des erstellten Symbols

Das Symbol **CD4081** ist nun vollständig definiert. Vergessen Sie nicht, dieses *Symbol* mit folgenden Kommandos zu *sichern*:

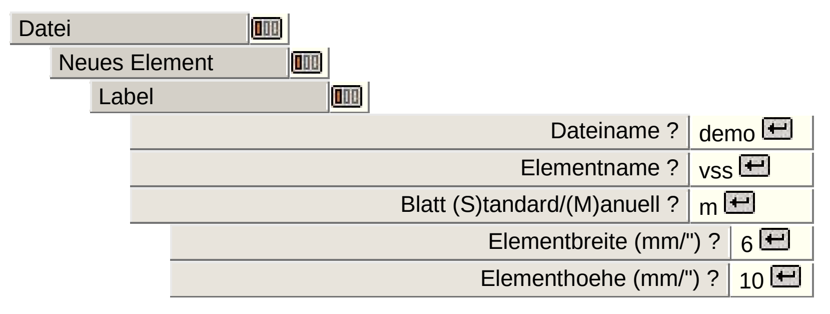


2.2.3 Label-Erstellung

Labelsymbole dienen auf Schaltplanebene der Definition von Signalen bzw. Netznamen. Mit Hilfe von Labels ist es daher möglich, Netze über verschiedene Stromlaufblätter einer Projektdatei hinweg miteinander zu verbinden. Labels werden auf Schaltplanebene mit der Funktion Neuer Label aus dem Menü Symbole geladen. Sofern ein Labelsymbol mit dem auf die Abfrage nach dem Netznamen spezifiziertem Namen verfügbar ist, wird dieses Symbol geladen. Ist dies nicht der Fall, dann wird (sofern in der eingestellten Bibliothek verfügbar) per Default das Labelsymbol **`standard`** geladen, welches eine Referenz zur Anzeige des Netznamens enthalten sollte. Neben dem Labelsymbol **`standard`** haben auch die beiden im System vordefinierten Labelsymbole **`bustap`** und **`port`** besondere Bedeutung. Das Labelsymbol **`bustap`** wird für Busanschlüsse verwendet. Das Labelsymbol **`port`** wird für die Definition von Modulports beim hierarchischen Schaltplandesign verwendet. Im Folgenden werden wir in der DDB-Datei **`demo.ddb`** die beiden Labelsymbole **`vss`** und **`vdd`** speziell zur Darstellung von Masse bzw. positiver Versorgung für CMOS-Technologie erstellen.

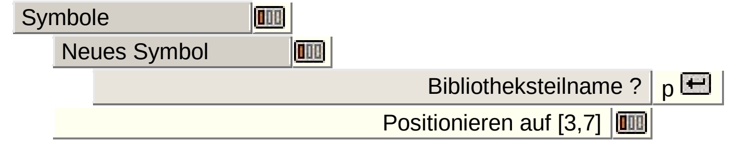
Erzeugen des Labelsymbols

Erzeugen Sie mit folgenden Kommandos ein neues Labelsymbol (d.h. ein neues Element vom Typ Label) mit dem Elementnamen **`vss`** innerhalb der Datei **`demo.ddb`** (sofern vom vorhergehenden Arbeitsschritt noch ein Element der Datei **`demo.ddb`** geladen ist, genügt die Selektion des Projekt-Buttons im Popupmenü bzw. die Angabe eines leeren Strings (Betätigen der Eingabetaste ↵) auf die Abfrage nach dem Dateinamen):



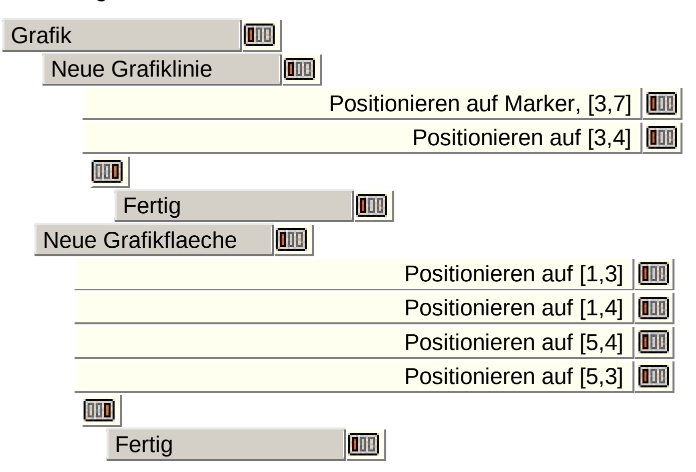
Platzieren des Pinsymbols

Platzieren Sie mit folgenden Kommandos das Markersymbol **`p`** an der Koordinate [3,7]:



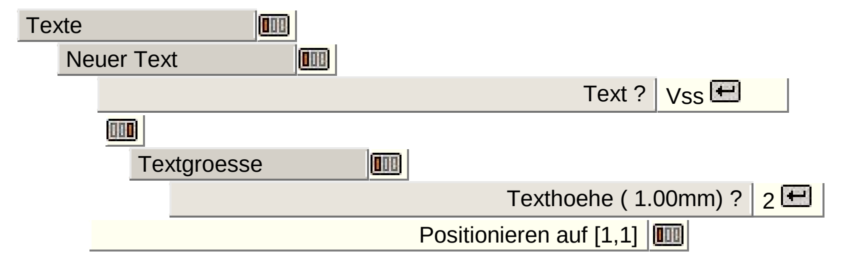
Grafik

Zeichnen Sie eine vertikale Grafiklinie ausgehend vom eben positionierten Marker bis zur Koordinate [3,4], und generieren Sie eine rechteckige Grafikfläche:



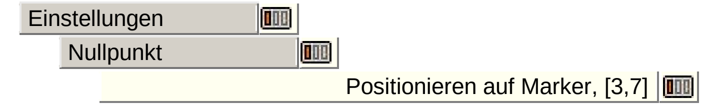
Text

Tragen Sie mit folgenden Kommandos den Label-Namen **Vss** als Text (Texthöhe 2mm) an der Koordinate [1,1] ein:



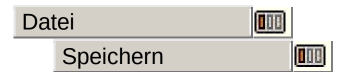
Definieren des Nullpunktes

Setzen Sie den Element-Nullpunkt des Labels auf die aktuellen Marker-Position (Koordinate [3,7]):



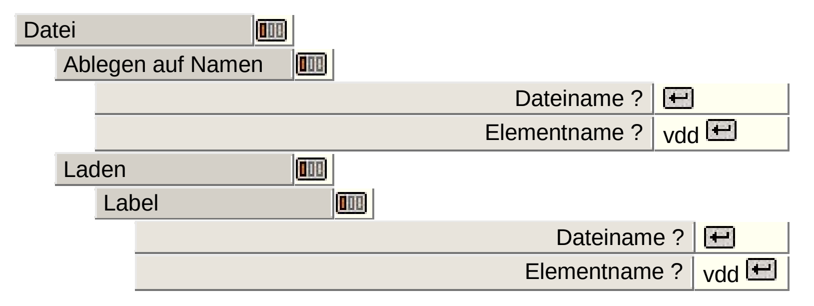
Sichern des erstellten Symbols

Das Labelsymbol **vss** ist jetzt vollständig definiert. Vergessen Sie nicht, dieses *Symbol* mit folgenden Kommandos abzu*speichern*:



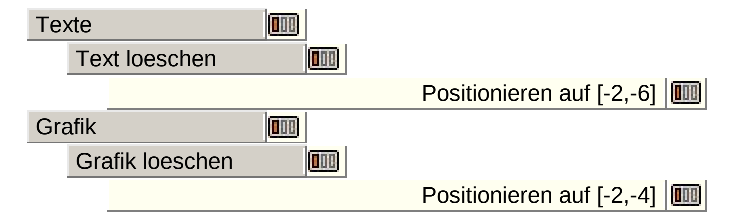
Erzeugen eines neuen Labelsymbols aus einem bestehenden

Das Labelsymbol **vdd** kann aus dem soeben erstellten Labelsymbol **vss** generiert werden. Kopieren Sie hierzu das noch geladene Labelsymbol **vss** auf **vdd**, und laden Sie **vdd**:

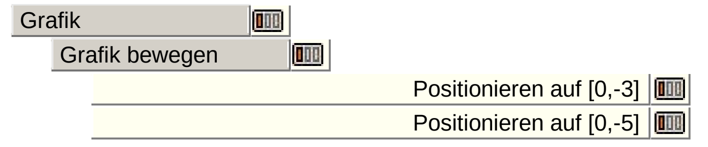


Im Grafikarbeitsbereich sehen Sie nun das Labelsymbol **vdd**, welches zunächst noch genauso aussieht, wie das Symbol **vss**. Durch einige Manipulationen können Sie das geladene Labelsymbol abändern.

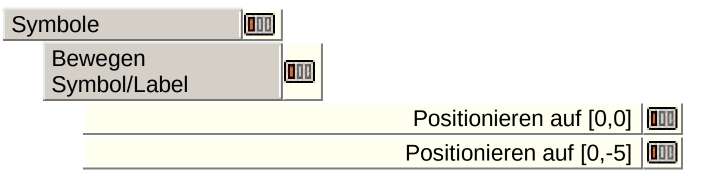
Löschen Sie zunächst mit den folgenden Kommandos den Text **Vss** sowie die rechteckige Grafikfläche:



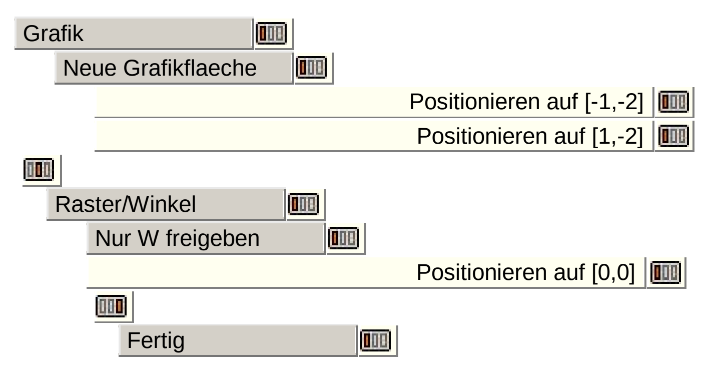
Bewegen Sie nun die vertikale Grafiklinie um 2mm nach unten:



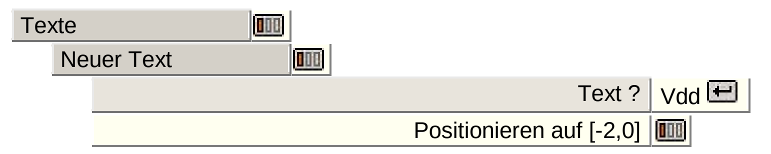
Bewegen Sie nun das Markersymbol um 5mm nach unten:



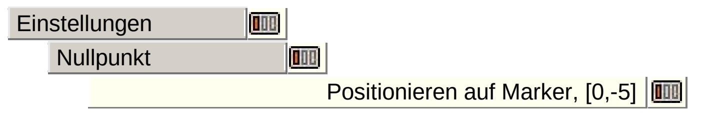
Generieren Sie nun eine dreieckige Grafikfläche, die in Form einer Pfeilspitze nach oben zeigt (um einen spitzen Winkel zu erzeugen, ist es notwendig, zwischenzeitlich in das Menü Ansicht zu wechseln, um den Winkel freizugeben):



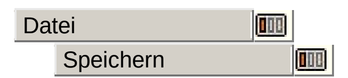
Tragen Sie mit folgenden Kommandos den Label-Namen **Vdd** als Text (Texthöhe 2mm ist noch eingestellt) an der Koordinate [-2,0] ein:



Setzen Sie den Elementnullpunkt des Labels auf die aktuelle Markerposition (Koordinate [0,-5]):



Das Labelsymbol **vdd** ist nun vollständig definiert. Vergessen Sie nicht, dieses *Symbol* mit folgenden Kommandos abzu*speichern*:

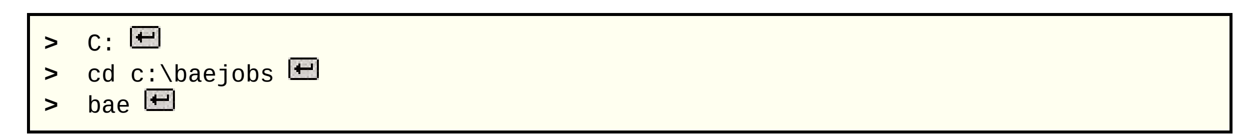


2.3 Schaltplanerstellung

In diesem Abschnitt wird anhand eines Beispiels die Schaltplanerstellung mit dem Stromlauf-Editor des **Bartels AutoEngineer** beschrieben. Dabei werden in der DDB-Datei **demo.ddb** zwei Stromlaufblätter mit den Elementnamen **sheet1** und **sheet2** editiert. **sheet1** wird die eigentliche Stromlauflogik enthalten, während auf **sheet2** die Steckerbelegung und Netzattribute zur Steuerung des **Autorouters** definiert werden.

Im Verlauf der Schaltplanbearbeitung werden wir uns vertraut machen mit den Standardfunktionen zur Generierung von Schaltplänen, zur Platzierung von Symbolen, zum Setzen von Attributwerten, zum Verlegen von Verbindungen, zum Platzieren von Labels, zum Definieren von Bussen, zum Definieren von Texten und zum Zeichnen von Grafik. Darüber hinaus werden wir auch spezielle Funktionen anwenden wie z.B. virtuelle Symboldefinitionen, Gruppenfunktionen, Definition von Steckerbelegung und Netzattributen, Verwendung von Templates, usw.

Wechseln Sie (sofern nicht schon geschehen) zunächst in das bei der Software-Installation angelegte BAE-Jobs-Verzeichnis (z.B. **c:\baejobs**) und starten Sie den **Bartels AutoEngineer**:



Rufen Sie das Schaltplan-Modul auf:

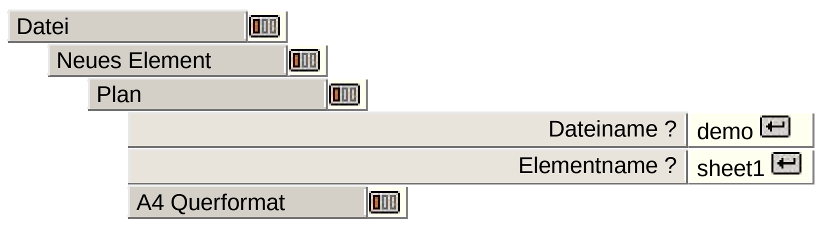


Sie befinden sich nun im Stromlauf-Editor des **Bartels AutoEngineer** und können mit der Erstellung von Stromlaufplänen beginnen.

2.3.1 Erstellen und Bearbeiten von Schaltplänen

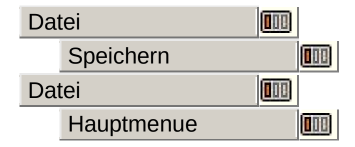
Erstellen eines neuen Schaltplans

Erzeugen Sie im DDB-File **`demo.ddb`** mit den folgenden Kommandos ein neues Stromlaufblatt mit dem Elementnamen **`sheet1`** und der Blattgröße A4:



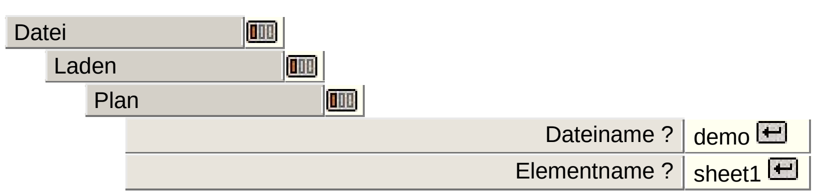
Nun sollte als heller Rahmen ein neuer, allerdings noch leerer Schaltplan auf dem Bildschirm erscheinen. Gibt das System z.B. die Fehlermeldung **`Dieses Element existiert bereits!`** aus, dann existiert der Plan **`sheet1`** schon in der Datei **`demo.ddb`**. In diesem Fall ist es nicht möglich, diesen Plan neu zu generieren, sondern er muss geladen werden (siehe unten). Die Abfrage nach dem Elementnamen des Stromlaufblattes ist natürlich notwendig, da ein Stromlaufplan aus mehr als einem Blatt bestehen kann. Bei der Vergabe dieses Namens haben Sie völlige Freiheit (Beispiele: **`blatt1`**, **`blatt2`**, **`plan_1`**, **`1`**, **`2`**, **`io`**, **`memory`**, **`steckerbelegung`**, usw.). Nach der Spezifikation des Elementnamens ist noch die Stromlaufblattgröße zu spezifizieren. Nach den Abfragen für den Datei- und Elementnamen wird ein Optionsmenü zur Selektion der Stromlaufblattgröße aktiviert. Hierbei stehen die vordefinierten Formate A0, A1, A2, A3, A4, Letter (8.5*11 Inch) und Tabloid (11*17 Inch) jeweils im Hoch- und Querformat zur Auswahl. Mit der Option Anderes Format kann die Blattgröße explizit über entsprechende Abfragen nach der Elementbreite und der Elementhöhe definiert werden.

Sie haben nun bereits einen neuen Schaltplan erzeugt. Speichern Sie diesen ab, und springen Sie zurück in die BAE-Shell:



Bearbeiten eines bestehenden Schaltplans

Rufen Sie nun erneut das Schaltplan-Paket auf, und laden Sie mit folgenden Kommandos den im vorhergehenden Arbeitsschritt erzeugten Schaltplan; da das Schaltplanelement bereits existiert, können sowohl der Dateiname als auch der Elementname wahlweise auch über das Popupmenü durch Mausklick selektiert werden:



Nun erscheint Ihr bereits erstelltes Stromlaufblatt auf dem Schirm. Beim Versuch, ein Element zu laden, meldet das System **`Datei nicht gefunden!`**, wenn die DDB-Datei nicht existiert, bzw. **`Plan nicht gefunden!`**, wenn das gewählte Element nicht in der DDB-Datei gefunden werden konnte.

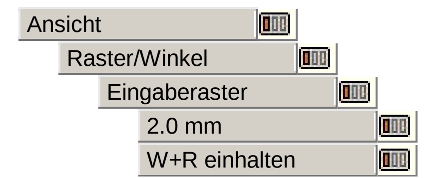
Beim Aufruf des **Schematic Editors** ist dem System der Dateiname des zuvor in einem anderen BAE-Programm-Modul bearbeiteten Elements bekannt. Die Spezifikation dieses systemweiten Projektnamens kann durch Selektion des Buttons Projekt im Dateinamens-Popupmenü oder durch die Eingabe eines Leerstrings (Betätigen der Eingabetaste ↵) auf die Abfrage nach dem Dateinamen erfolgen.

2.3.2 Symbole

Das Menü Symbole bietet die Möglichkeit, aus unterschiedlichen, selektierbaren Schaltzeichenbibliotheken Stromlaufsymbole (und Labels) in das Schema zu laden und diese auf dem Schaltplan zu platzieren. Auch können einmal platzierte Symbole wieder gelöscht werden. Die im Menü Symbole ausgeführten Arbeitsschritte erzeugen also im wesentlichen die Bauteilliste des Designs.

Eingaberaster

Grundsätzlich können im **Bartels AutoEngineer** alle Grafikeingaben in beliebigen Rastern oder auch rasterfrei erfolgen. Dennoch sollte die Platzierung von Symbolen in einem definierten Raster erfolgen, um sicherzustellen, dass später beim Verlegen der Verbindungen in einem vernünftigen Eingaberaster gearbeitet werden kann. Stellen Sie also zunächst mit folgenden Kommandos ein Eingaberaster von 2mm ein, das grob genug ist, um ohne Probleme später im feineren 1mm-Raster Verbindungen an die Symbolanschlüsse zu legen:



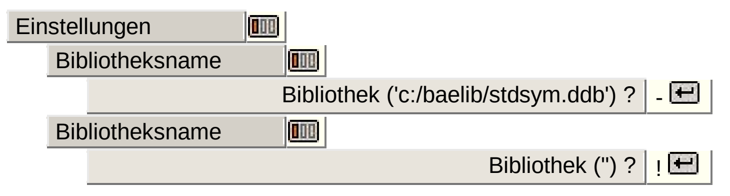
Bibliothekszugriff

Die Funktion Bibliotheksname im Menü Einstellungen ermöglicht die Selektion der Schaltzeichenbibliothek, aus der die Symbole geladen werden sollen. Überprüfen Sie zunächst mit folgenden Kommandos, wie der Bibliothekspfad gesetzt ist:



Im Prompt zur Abfrage nach der Bibliothek zeigt das System den Namen der aktuell eingestellten Bibliothek an. Nach dem Aufruf des **Schematic Editors** ist dies zunächst der Name der über das Setup eingestellten SCM-Standardbibliothek (siehe hierzu die Beschreibung des Utilityprogramms **BSETUP** im Kapitel 7.2 dieses Handbuchs).

Durch die Eingabe eines leeren Strings auf die Abfrage nach der Bibliothek ändert sich der Bibliotheksname nicht. Durch die Eingabe von **-** wird der Bibliotheksname zurückgesetzt, d.h. es ist dann keine Bibliothek selektiert. Die Eingabe von **!** bzw. **.** setzt den Bibliotheksnamen wieder auf die durch das Setup eingestellte Bibliothek. Überprüfen Sie dies mit folgenden Kommandos:



In den Windowsversionen der BAE-Software erfolgt die Abfrage des Bibliotheksnamens über einen Dateiabfragedialog.

Einige grundlegende Anmerkungen zum Thema Bibliotheken. Bekanntlich ist eines der Leistungsmerkmale des **AutoEngineer** das flexible Datenbankkonzept. Dieses Konzept impliziert, dass auch jede Projektdatei als Bibliothek fungieren und als solche im System angemeldet werden kann. Aktiviert der Anwender die Funktion zum Laden eines Symbols, dann sucht das System zunächst innerhalb der aktuellen Projektdatei nach dem angeforderten Bibliotheksteil. Ist das Element nicht hierin schon enthalten, wird die Suche in der im System angemeldeten Standardbibliothek fortgesetzt. Immer, wenn ein Symbol aus einer Bibliothek in das Schema geladen wird, erstellt das System automatisch eine Kopie des Symbols in der aktuellen Projektdatei. Das Symbol ist dann in der aktuellen Projektdatei abgespeichert, wird also bei mehrfachem Platzieren nicht mehr aus der entsprechenden Bibliothek geholt. Abbildung 2-3 verdeutlicht das Schema des Bibliothekszugriffs innerhalb des **Schematic Editors**.

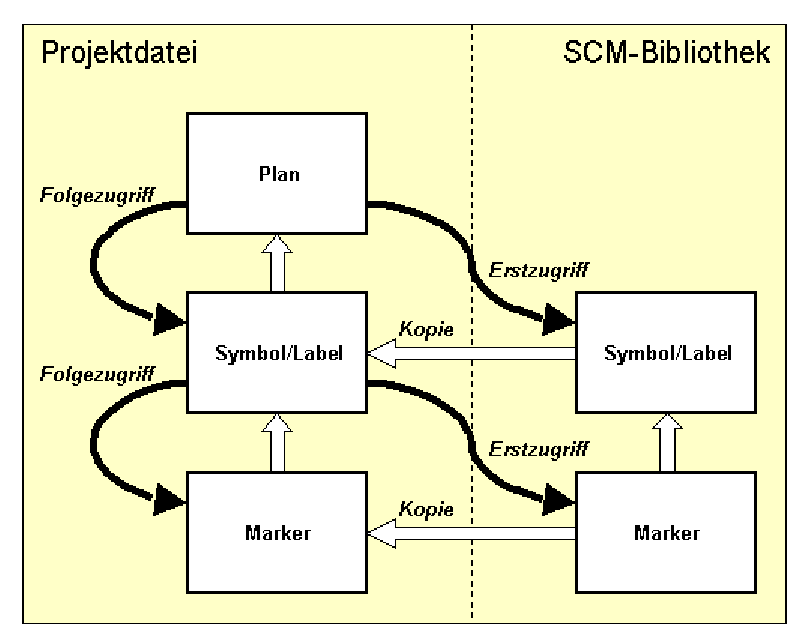
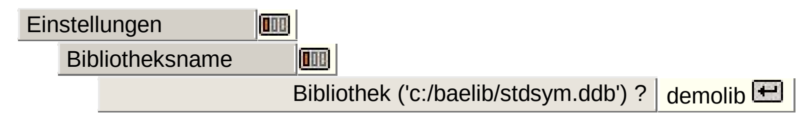


Abbildung 2-3: SCM-Bibliothekszugriff

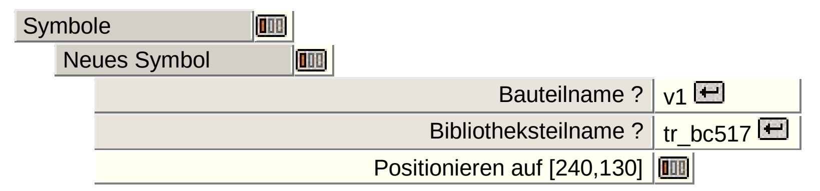
Das Umschalten der Default-Bibliothek innerhalb des Systems ist im Grunde nur dann sinnvoll, wenn eine Reihe verschiedener Symbole aus einer nicht über den Bibliothekspfad erreichbaren Bibliothek zum ersten Mal im Schema platziert werden sollen, wie in unserem Fall z.B. aus der Bibliotheksdatei **`demolib.ddb`** im aktuellen Verzeichnis. Stellen Sie nun den Bibliothekspfad auf **`demolib.ddb`** ein:



Durch obige Eingabe ist die DDB-Datei **`demolib.ddb`** des Arbeitsverzeichnisses als aktuelle Symbolbibliothek im System angemeldet, und es können nun Symbole aus dieser Bibliothek in den Schaltplan geladen werden.

Laden von Symbolen

Laden Sie mit folgenden Kommandos das Transistorsymbol **tr_bc517**, geben Sie diesem Bauteil den Namen **V1** und positionieren Sie es an der Koordinate [240,130]:

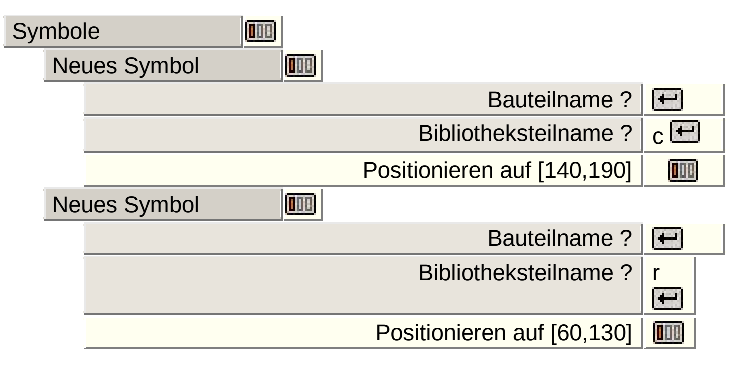


Auf die Abfrage nach dem Bauteilnamen ist die Referenz des Symbols (z.B. **IC10**, **R8**, **U004**, ...) einzugeben. Wird hierbei ein leerer String eingegeben (Betätigen der Eingabetaste ↵), dann generiert das System den Bauteilnamen entsprechend dem für das Symbol definierten Symbolnamensmuster automatisch (siehe hierzu auch Kapitel 2.2.2, Symbolerstellung). Diese Leerstringeingabe kann wahlweise auch durch die Betätigung einer beliebigen Maustaste aktiviert werden. Das System gibt eine Fehlermeldung aus, wenn der über das Bauteilnamensmuster definierte Namensraum ausgeschöpft ist, d.h. wenn kein dem Namensmuster entsprechender neuer Bauteilname mehr automatisch erzeugt werden kann.

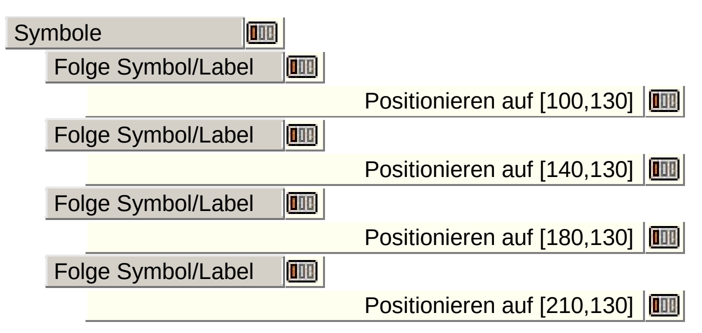
Die in einem Schaltplan verwendeten Bauteilnamen müssen eindeutig sein, damit bei späteren **Packager**- oder **Backannotation**-Läufen keine Konflikte entstehen können. Bei Verwendung der automatischen Bauteilbenennung ist sichergestellt, dass die Bauteilnamen eindeutig sind. Um Namenskonflikte bei der expliziten Bauteilbenennung zu vermeiden, überprüft das System, ob der angegebene Bauteilname bereits im Schaltplan verwendet wird. Ist dies der Fall, dann erfolgt eine entsprechende Bestätigungsabfrage, über die der Anwender angeben muss, ob der gewünschte Bauteilname tatsächlich verwendet werden soll. Befindet sich ein Bauteil mit dem gleichen Namen auf dem aktuell bearbeiteten Stromlaufblatt so wird dieses gelöscht und durch das neue Bauteil (ggf. mit anderem Symbol, in jedem Fall jedoch mit uninitialisierten Attributwerteinträgen) ersetzt. Befindet sich hingegen ein Bauteil mit dem gleichen Namen auf einem anderen als dem aktuell geladenen Stromlaufblatt des Stromlaufplans, so wird dieses nicht automatisch entfernt, sondern der Anwender muss dieses dann explizit aus dem Schaltplan löschen, um Namenskonflikte in nachfolgenden **Packager**läufen zu vermeiden.

Auf die Abfrage nach dem Bibliotheksteilnamen erwartet das System den Namen des Symbols innerhalb der Bibliothek. Die Betätigung einer beliebigen Maustaste bzw. die Eingabe eines Leerstrings oder eines Fragezeichens (**?**) bewirkt an dieser Stelle die Aktivierung eines Popupmenüs mit allen im Bibliotheksverzeichnis verfügbaren Bibliotheksdateien. Hierbei werden auch die Buttons Bibl. und Projekt angezeigt. Mit Bibl. (oder durch Eingabe von **>**) wird die aktuell über Bibliotheksname aus dem Menü Einstellungen selektierte Standardbibliothek ausgewählt. Mit Projekt ist der Zugriff auf die Stromlaufsymbole der aktuellen DDB-Datei, d.h. auf die Projektdatenbank möglich. Nach der Auswahl einer der angebotenen Bibliotheksdateien erfolgt die Aktivierung eines weiteren Popupmenüs mit der Liste der in der selektierten Bibliothek verfügbaren Symbole.

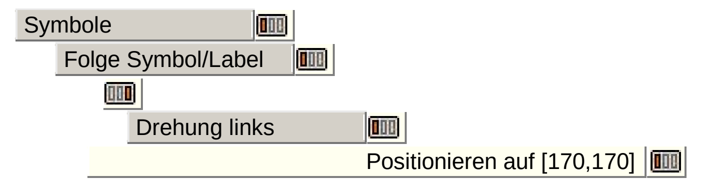
Laden Sie mit den folgenden Kommandos das Kondensatorsymbol **c** und das Widerstandssymbol **r** in Ihr Schema (jeweils mit automatisch generiertem Bauteilnamen):



Die Funktion Folgesymbol sorgt dafür, dass dasselbe Symbol, wie beim vorhergehenden Ladevorgang mit automatisch erzeugtem Bauteilnamen geladen wird. Platzieren Sie mit den folgenden Kommandos drei weitere Widerstandssymbole auf dem Schaltplan:



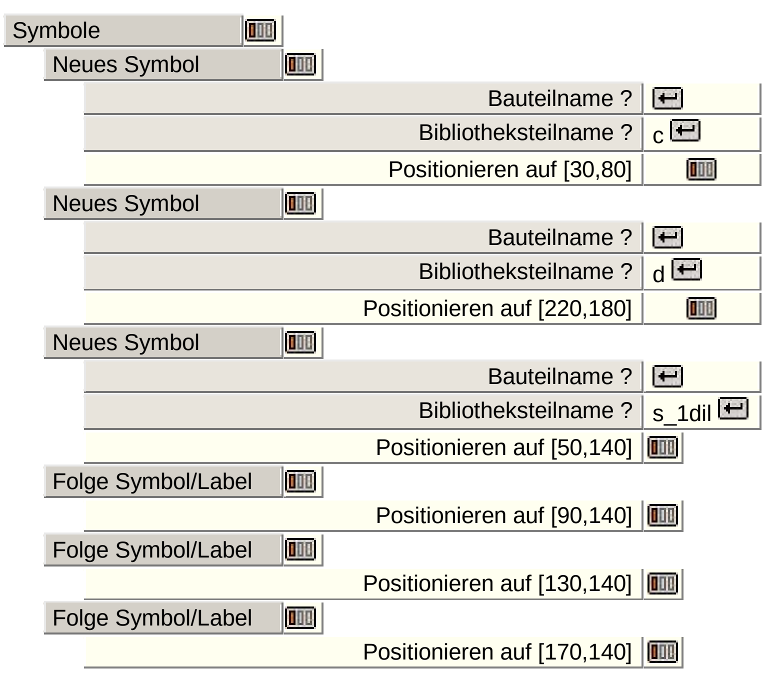
Das Untermenü, das während des Platzierens des Symbols über die rechte Maustaste aufgerufen werden kann, dient dazu, das Symbol je nach Bedarf zu drehen, zu spiegeln, oder absolut zu platzieren. Laden Sie ein weiteres Widerstandssymbol, drehen Sie es, und platzieren Sie es an der Koordinate [170,170]:



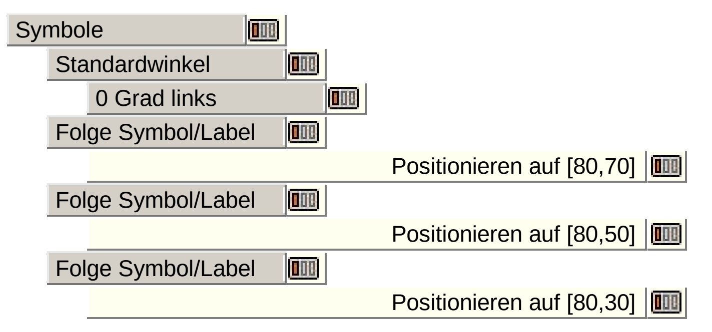
Wenn nacheinander mehrere Symbole um einen bestimmten Winkel gedreht platziert werden sollen, dann empfiehlt es sich, den Standardwinkel z.B. wie folgt einzustellen:



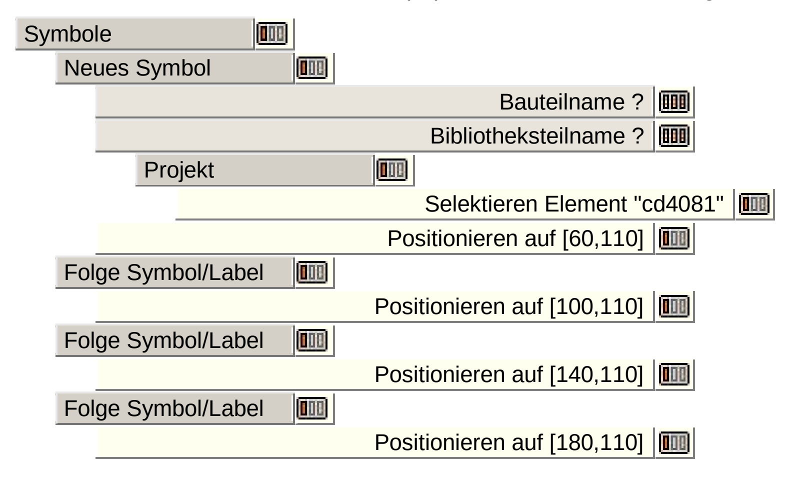
Obige Einstellung des Standardwinkels bewirkt, dass nun jedes neu zu platzierende Symbol automatisch um 90 Grad gedreht geladen wird. Überprüfen Sie dies, indem Sie einige weitere Symbole laden:



Setzen Sie nun wieder den Standardwinkel für die Platzierung auf 0 Grad, und laden Sie einige weitere Symbole:

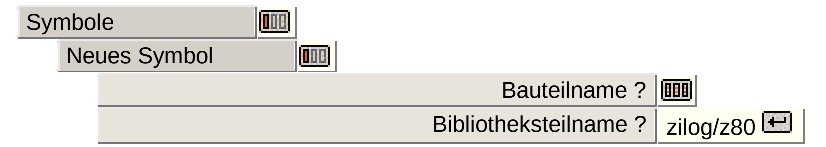


Einmal in eine Projektdatei geladene Symbole sind direkt verfügbar. Das bedeutet auch, dass wir das Bibliothekssymbol CD4081, welches wir in Kapitel 2.2.2 erstellt und in unserer Projektdatei **`demo.ddb`** abgelegt haben, ungeachtet des eingestellten Bibliothekspfades mehrfach auf dem aktuellen Schaltplanblatt platzieren können. Im folgenden Arbeitsschritt bedienen wir uns außerdem der Möglichkeit der Spezifikation des Bauteilnamens sowie der Selektion des Bibliothekselements über Mausklick bzw. Popupmenü, d.h. wir kommen gänzlich ohne Tastatureingaben aus:

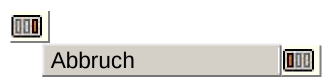


Wie Sie in obigem Arbeitsschritt gesehen haben, bietet Ihnen das System über das entsprechende Popupmenü den Zugriff auf alle Bibliotheksdateien des Bibliothekspfades sowie auf die über den Button Projekt selektierbare projektspezifische Bibliothek. Mit den nachfolgend aktivierten Popupmenüs zur Symbolauswahl haben Sie gleichzeitig Zugriff auf ein Inhaltsverzeichnis der SCM-Symbolbibliothek (die Ladefunktion lässt sich ggf. durch Selektion des Buttons Abbruch abbrechen).

Sie können auch, während Sie eine Bibliothek im System angemeldet haben, Symbole aus einer völlig anderen Bibliothek laden. Laden Sie mit folgenden Kommandos das Symbol **`z80`** aus der Bibliothek **`zilog.ddb`**:



Nun sollte das Symbol **`z80`** am Fadenkreuz hängen und könnte platziert werden. Brechen Sie diesen Arbeitsschritt mit folgenden Kommandos ab:



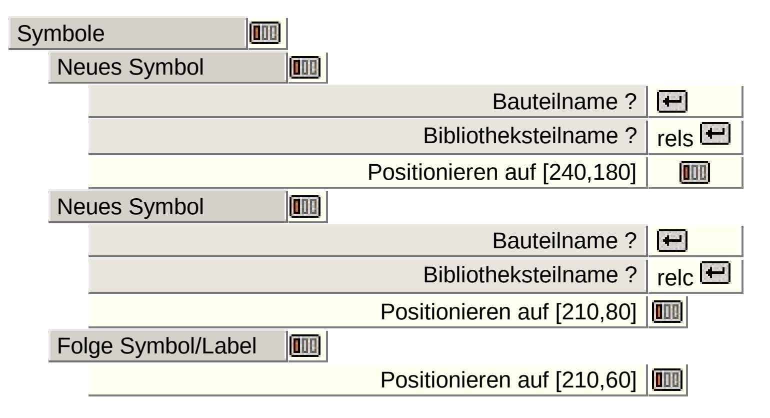
Obiger Abbruch während des Ladens bzw. Platzierens eines neuen, noch nicht in der aktuellen Projektdatei vorhandenen Symbols bewirkt, dass weder die Bauteilliste geändert wird, noch das Symbol selbst unnötigerweise in die Projektdatei kopiert wird.

Wie Sie weiterhin aus obigem Beispiel erkennen, ist die Syntax zur Spezifikation des Bibliotheksteilnamens gegeben durch:

```
<bibliotheksname>/<bibliotheksteilname>
```

Die Betätigung einer beliebige Maustaste oder die Angabe eines Fragezeichens **`?`**) aktiviert hierbei das Popupmenü zur Auswahl der Bibliotheksdatei. Die Angabe eines Fragezeichens anstelle des Bibliotheksteilnamens bewirkt die Aktivierung des Popupmenüs zur Auswahl der in der angegebenen Bibliotheksdatei verfügbaren Symbole (in unserem Beispiel also ein Popupmenü mit den in **`zilog.ddb`** enthaltenen Symbole auf die Eingabe **`zilog/?`**).

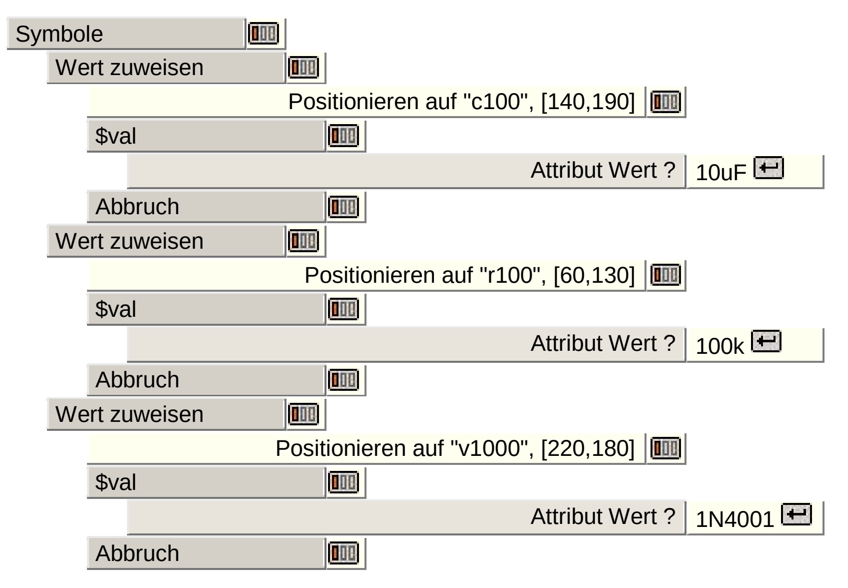
Im **Bartels AutoEngineer** besteht die Möglichkeit, *verschiedene Stromlaufsymbole* an *ein* und dasselbe *Gehäuse* zuzuweisen (siehe hierzu auch **Packager** und **LOGLIB**). Platzieren Sie mit folgenden Kommandos eine Spule und zwei Kontakte:



Die in obigem Arbeitsschritt platzierten Symbole sind per Bibliotheksdefinition (siehe auch **LOGLIB**-Datei **`demolib.def`**) einem Relais-Bauteil zugeordnet, und werden später vom **Packager** entsprechend in *ein einziges* Gehäuse gepackt.

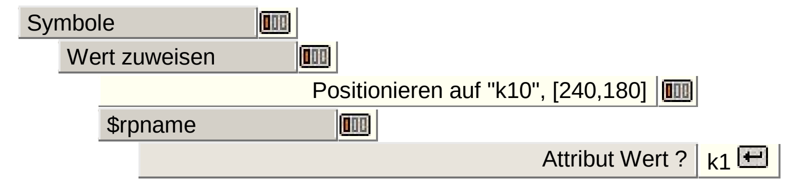
Attributwertzuweisungen

Sofern auf Symbolebene Attribute definiert wurden, können nun auf Planebene entsprechende Attributwerte zugewiesen werden. Mit folgenden Kommandos kann das an den diskreten Bauteilen **`c100`**, **`r100`** und **`v1000`** definierte Attribut **`$val`** jeweils mit einem Attributwert versehen werden:



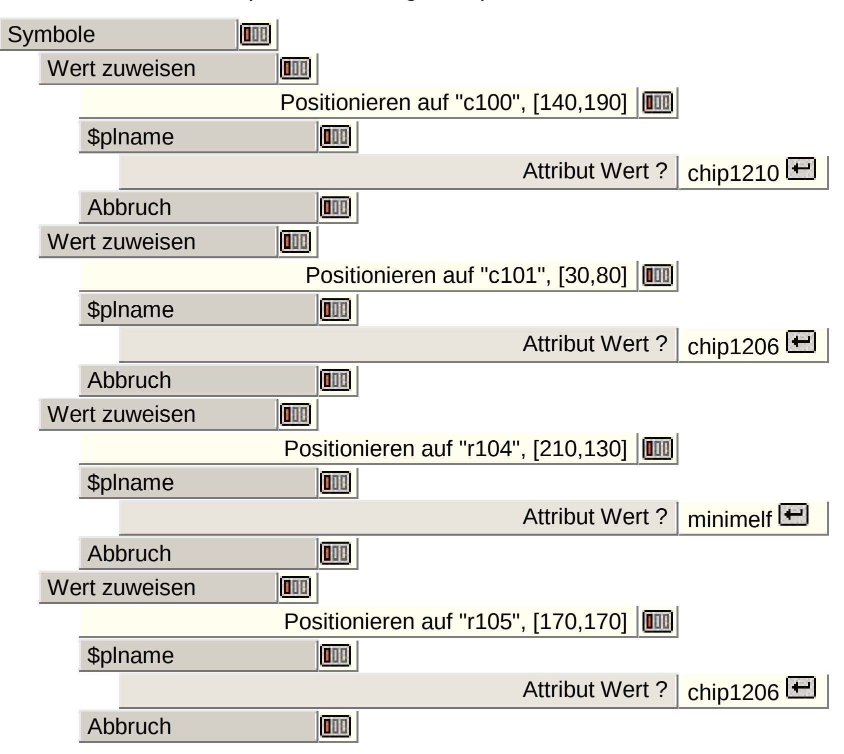
Nehmen Sie auf dieselbe Weise auch entsprechende Werteinträge für das Attribut **`$val`** der Bauteile **`r101`**, **`r102`**, **`r103`** (jeweils **`100k`**), **`r104`** (**`10k`**), **`r105`** (**`1M`**) und **`c101`** (**`100pF`**) vor. Beachten Sie, dass zur besseren Identifikation des selektierten Symbols während der Abfrage nach dem Attributnamen in der Statuszeile jeweils der Bauteil- und Symbolname angezeigt werden (z.B. **`Log. Name: c100, Macro: c`**).

Das Untermenü zur Auswahl des Attributes zeigt die im jeweiligen Symbol definierten Attribute an, d.h. dieses Menü kann je nach selektiertem Symbol unterschiedlich aufgebaut sein. Die Option Abbruch wird hierbei nur dann angezeigt und muss zur Beendigung der Funktion angewählt werden, wenn am selektierten Symbol mehr als ein Attribut definiert ist. An den Symbolen **rels** und **relc** z.B. ist jeweils nur das Attribut **$rpname** definiert. Um den Packager anzuweisen, das Symbol **k10** in das Bauteil mit dem Namen **k1** (bei Beibehaltung der definierten Gehäusezuweisung) zu packen, ist folgende Attributwertzuweisung vorzunehmen:



Nehmen Sie dieselbe Attributwertzuweisung (Wert **k1** an Attribut **$rpname**) auch für die Bauteile **KK100** und **KK101** (Relaiskontakte **relc**) vor, um den **Packager** zu veranlassen, diese beiden Symbole zusammen mit der Relaisspule **K10** in das Bauteil **K1** zu packen.

Mit Hilfe des Attributes **$plname** kann der **Packager** veranlasst werden, ein Stromlaufsymbol abweichend von der in der Bibliothek eingetragenen Default-Definition einem anderen Gehäusetyp, d.h. einem anderen Layoutsymbol zuzuweisen. Führen Sie mit den folgenden Kommandos jeweils Wertzuweisungen für das Attribut **$plname** der Bauteile **c100**, **c101**, **r104** und **r105**, um diese Bauteile durch den **Packager** an entsprechende SMD-Gehäusebauformen (anstelle bedrahteter Gehäuse, wie in der Bibliothek per Default eingestellt) zuzuweisen:



Haben Sie alle Arbeitsschritte dieses Abschnitts korrekt ausgeführt, dann sollten Sie auf Ihrem Bildschirm nun das in Abbildung 2-4 gezeigte Stromlaufbild sehen.

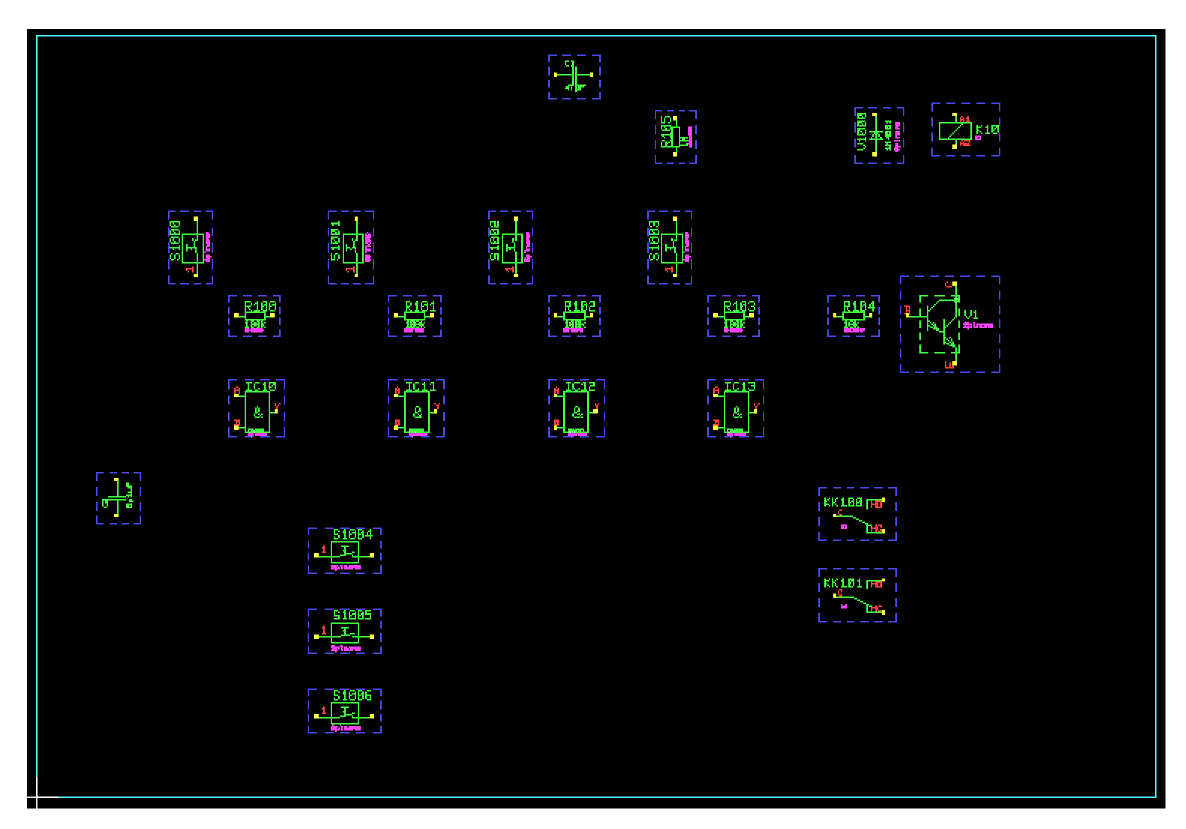


Abbildung 2-4: Stromlauf mit platzierten Symbolen

Symbole bewegen, Symbole löschen, Symbolnamen ändern

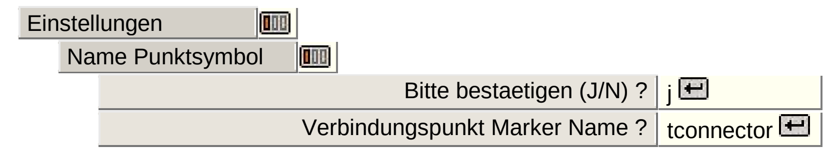
Mit Hilfe der Funktion Bewegen Symbol/Label können bereits platzierte Symbole bewegt, gedreht und gespiegelt werden (die rechte Maustaste aktiviert ein entsprechendes Untermenü). Im **Schaltplaneditor** ist ein Verfahren zum automatischen Neuverlegen der Verbindungen zu einem umplatzierten Symbol bzw. Label integriert (Signalrouting, siehe unten). Das Löschen eines platzierten Symbols erfolgt mit der Funktion Loeschen Symbol/Label. Bauteilnamen bzw. Referenzen können mit der Funktion Symbolname aendern geändert werden. Testen Sie diese Funktionen auf Ihrem aktuellen Schaltplanblatt, und bedienen Sie sich hierbei auch der Funktionen Undo und Redo, um Realisierungsalternativen durchzuspielen. Beim Symbolpick ist zu beachten, dass das Symbol an der Pickposition, das sich komplett innerhalb der Grenzen anderer Symbole an der Pickposition befindet, auf jeden Fall Priorität beim Pick hat, um eine unbeabsichtigte Selektion von Rahmensymbolen zu verhindern.

2.3.3 Verbindungen, Labels, Busse

Das Menü Verbindungen dient dazu, Verbindungen in den Schaltplan einzuzeichnen sowie Busse zu definieren und anzuschließen. Auch können von hier aus Verbindungssegmente bewegt, geteilt oder gelöscht werden.

Selektion des Verbindungspunkt-Markers

Mit folgenden Kommandos kann das Markersymbol **tconnector**, das zur Kennzeichnungen von T-förmigen Verbindungsstücken verwendet werden soll, selektiert werden:



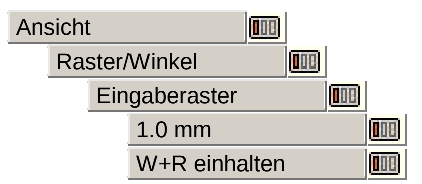
Der Verbindungspunkt-Marker ist ein spezielles Markersymbol. Im Gegensatz zum Pin-Markersymbol sollte der Verbindungspunkt-Marker eine normale Grafikfläche anstelle des Kontaktbereiches enthalten. Per Default wird vom System der Verbindungspunkt-Marker **junction** verwendet (achten Sie darauf, dass dieser Marker auch in der voreingestellten Bibliothek verfügbar ist; siehe hierzu auch die Beschreibung des Kommandos **SCMDEFLIBRARY** im Programm **BSETUP**).

Grafikkontrollfunktion, Eingaberaster

Wird eine Verbindung an einen Pin angeschlossen, dann wird der am Pinsymbol definierte Kontaktbereich unsichtbar (siehe hierzu auch 2.2.1, Marker-Erstellung), und die entsprechende Änderung in der Netzliste ist somit grafisch angezeigt. Da der **AutoEngineer** die Möglichkeit bietet, Verbindungen "im Leeren" enden zu lassen (dies ist z.B. bei der Definition von Bussen notwendig, siehe unten), ist diese wichtige Kontrollfunktion dringend zu beachten, um sicherzustellen, dass die richtige Netzliste generiert wird. Beachten Sie, dass bei Anschluss eines *Einzel*segments ohne Anschluss zu einem anderen Pin der Kontaktbereich ebenfalls ausgeblendet wird. Damit verfügt der Anwender über eine Kontrollfunktion zum "Abhaken" bearbeiteter Pins, d.h. mit dieser Funktion kann die farbliche Hervorhebung in einfacher Weise für solche Pins zurückgenommen werden, die mit keinem anderen Pin zu verbinden sind (also vom Entwickler im Folgenden auch nicht mehr gesondert zu betrachten sind). Im Gegensatz hierzu werden Kontaktbereiche als Fehler durch Anzeige in Highlightfarbe gekennzeichnet, wenn eine offene Verbindung bestehend aus *mehreren* Segmenten am Pin angeschlossen ist. Dieser Spezialfall wird in der Reportanzeige (Funktion Report aus dem Menü Utilities) zusätzlich als Zeichenfehler gezählt.

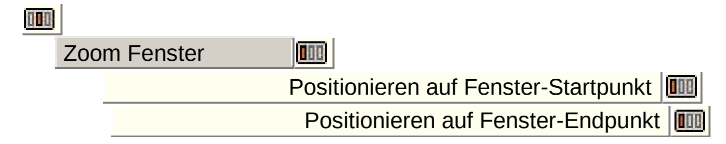
Sollte es Ihnen im eingestellten Eingaberaster nicht gelingen, einen Pin anzuschließen (weil dieser in einem zu feinen Raster liegt), dann müssen Sie auf ein feineres Eingaberaster schalten oder u.U. das Eingaberaster über das Menü Ansicht freigeben.

Stellen Sie mit folgenden Kommandos das Eingaberaster auf 1mm ein, so dass Sie in den nachfolgenden Arbeitsschritten ohne Probleme die auf Symbolebene platzierten Symbolanschlüsse anschließen können:

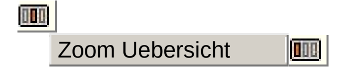


Ansicht, Bilddarstellung

Erinnern Sie sich bitte an die Bedeutung der mittleren Maustaste. Das über die mittlere Maustaste aufrufbare Menü Ansicht lässt sich *während* des Platzierens und dem Zeichnen von Geometrie und Verbindungen jederzeit aktivieren. Insbesondere die Zoomfunktionen sind sehr hilfreich bei der Generierung von Verbindungen. Mit folgenden Kommandos können Sie jederzeit in das Menü Ansicht gelangen und ein definiertes Fenster herauszoomen:

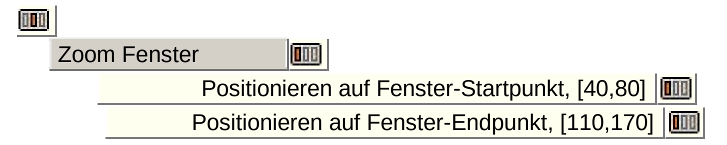


Um wieder in die Übersichtsdarstellung zu schalten, sind folgende Kommandos zu aktivieren:

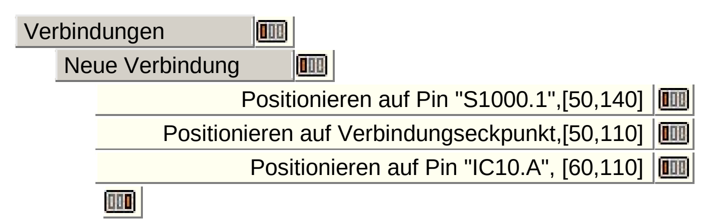


Verlegen von Verbindungen

Wir wollen nun eine erste Verbindung zwischen den Bauteilen **`S1000`** und **`IC10`** herstellen. Zoomen Sie hierzu zunächst in den Arbeitsbereich um diese beiden Symbole:

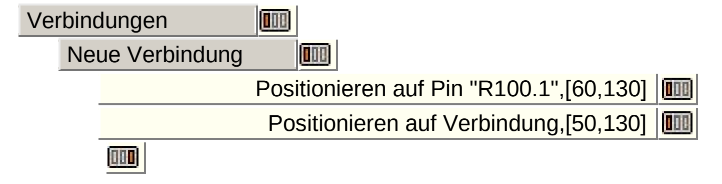


Verbinden Sie nun mit den folgenden Kommandos den Anschluss **`1`** des Bauteils **`S1000`** mit dem Anschluss **`A`** des Bauteils **`IC10`**:



Verbindungseckpunkte werden immer durch Betätigung von linken Maustaste gesetzt. Eine Verbindung wird durch Betätigung von der rechten Maustaste beendet. Nach Ausführung des obigen Arbeitsschrittes sollten die Anschlussflächen der beiden zu verbindenden Pins unsichtbar geworden sein. Ist dies nicht der Fall, dann machen Sie den Arbeitsschritt mit Hilfe der Undo-Funktion rückgängig, und versuchen Sie erneut, die gewünschte Verbindung (wenn nötig in einem feineren Eingaberaster) herzustellen.

Schließen Sie nun mit den folgenden Kommandos den Pin **`1`** des Widerstands **`R100`** an die soeben gelegte Verbindung an:



Wird haben nun eine T-förmige Verbindung zur zuvor gelegten Verbindung generiert. Das System sollte dieses T-Stück durch den Verbindungspunkt-Marker kennzeichnen.

Zeichnen Sie nun weitere Verbindungen ein, so dass sich das in Abbildung 2-5 dargestellte Schaltbild ergibt. Experimentieren Sie dabei auch mit den Funktionen zum Bewegen, Teilen und Löschen von Segmenten bzw. zum Löschen von Verbindungen oder ganzen Netzen. Bedienen Sie sich dabei der Undo/Redo-Funktionen, um Realisierungsalternativen durchzuspielen. Beachten Sie bitte insbesondere auch die Funktion Punkt zu Punkt. Damit wird automatisch eine Verbindung zwischen zwei selektierbaren Punkten auf dem aktuell geladenen Schaltplan gelegt, sofern dies mit maximal drei Verbindungssegmenten möglich ist.

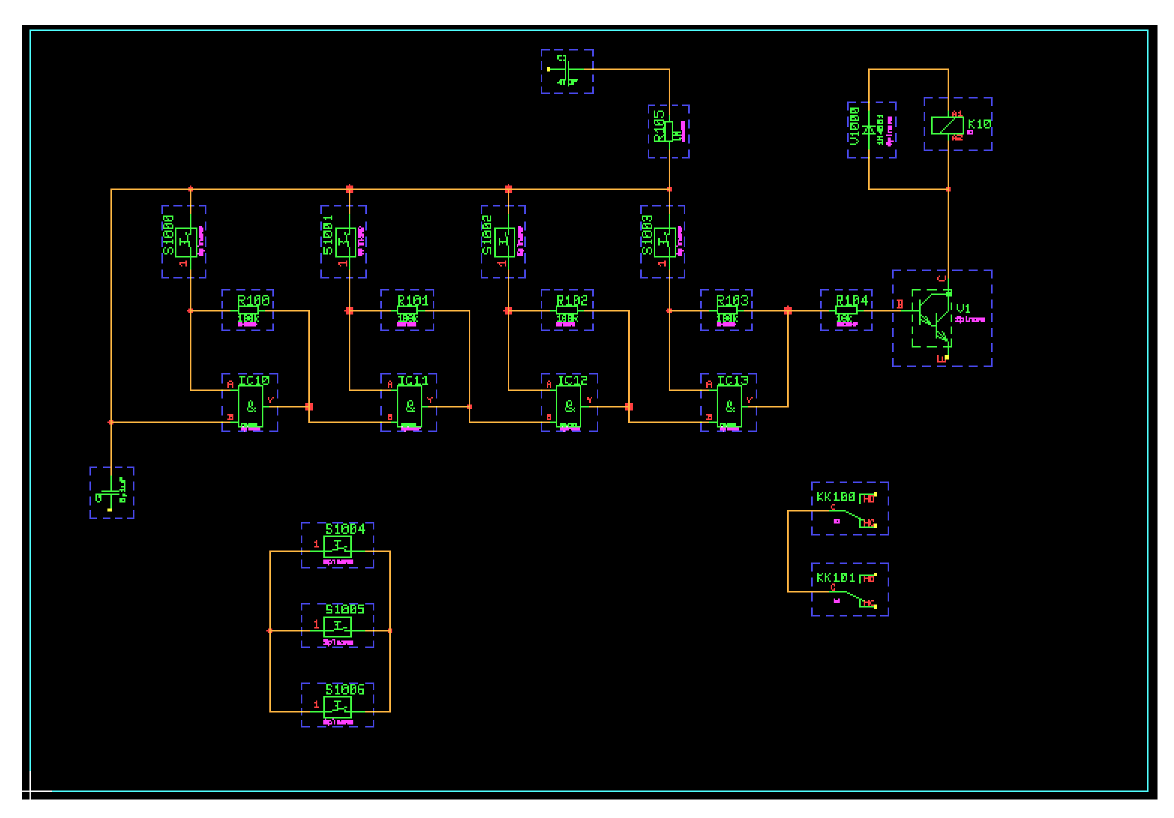


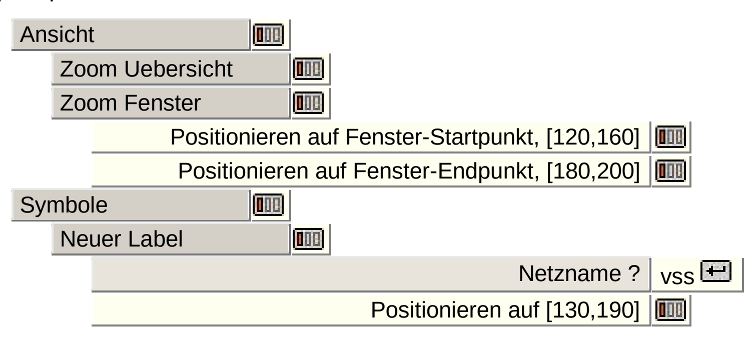
Abbildung 2-5: Stromlauf mit Symbolen und Verbindungen

Labels

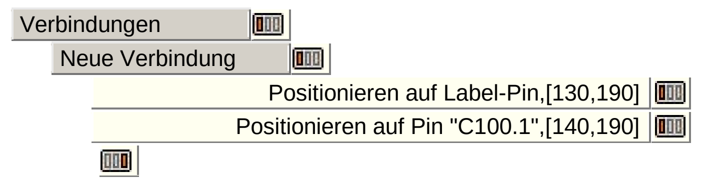
Labels sind spezielle Symbole zur Definition von Netznamen. Mit Hilfe von Labels ist es möglich, Verbindungen (auch stromlaufblattübergreifend) auf definierte Signale bzw. Signalpegel zu legen.

In Kapitel 2.2.3 haben wir die beiden Labelsymbole **vss** (0V-CMOS-Versorgung) und **vdd** (positive CMOS-Versorgung) in unserer Projektdatei **demo.ddb** erstellt. Diese beiden Symbole sind also in der aktuellen Projektdatei direkt verfügbar und können über das Menü Symbole in den Schaltplan geladen werden.

Definieren Sie mit den folgenden Kommandos das Signal **Vss** indem Sie den Label **vss** laden und diesen auf dem Schaltplan platzieren:

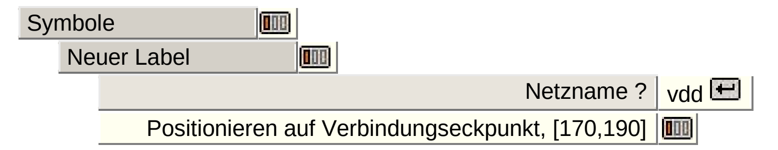


Erzeugen Sie mit den folgenden Kommandos einen Anschluss vom Pin **1** des Kondensators **C100** zum im vorhergehenden Arbeitsschritt definierten Signal **Vss**, indem Sie eine entsprechende Verbindung vom Kondensatoranschluss zum zuvor platzierten Label erzeugen:



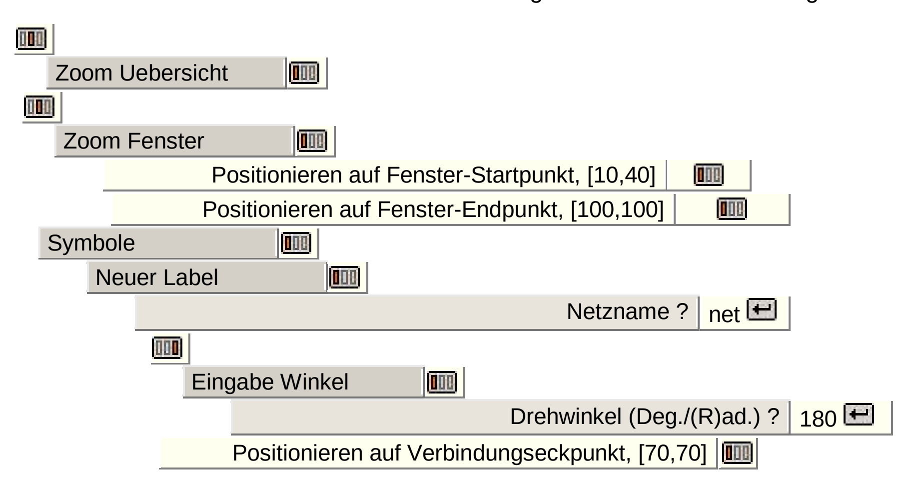
Achten Sie darauf, dass der am Pin des Labelsymbols definierte Kontaktbereich erlischt, um sicherzustellen, dass der Kondensator auch tatsächlich an **Vss** angeschlossen ist.

Es ist nicht nötig, explizit eine Verbindung zum Labelsymbol einzuzeichnen, wenn der Label auf den Eckpunkt einer anzuschließenden Verbindung gelegt wird. Laden Sie mit den folgenden Kommandos den Label **Vdd**, und schließen Sie diesen Label an die Verbindung zwischen dem Kondensator **C100** und **R105** an:



Achten Sie wieder darauf, dass der Kontaktbereich am Pin des Labelsymbols erlischt.

Ist kein spezielles Symbol mit dem entsprechenden Netznamen für einen zu ladenden Label definiert bzw. verfügbar, dann verwendet das System das Labelsymbol **`standard`** (dies setzt allerdings voraus, dass zumindest dieses Labelsymbol in der voreingestellten Bibliothek bzw. in der Projektdatei verfügbar ist). Definieren Sie mit folgenden Kommandos den Netznamen der an Pin **`1`** des Schalters **`S1004`** angeschlossenen Verbindung zu **`NET`**:



Die Verwendung verschiedener Labels auf einem Blatt unabhängig vom Labelnamen (Netznamen) kann über den Parameter Name Labelmakro gesteuert werden. Beim Platzieren eines Labels zu dem kein gleichnamiges Labelsymbol gefunden wird, wird das über diesen Parameter spezifizierte Makro verwendet. Der neue Parameter kann über die Funktion Einstellungen aus dem Menü Einstellungen gesetzt werden. Damit ist es möglich, bestimmte Netztypen durch unterschiedliche Labels zu kennzeichnen, ohne dass für jeden einzelnen Netznamen ein spezielles Labelsymbol erstellt werden müsste. Als Voreinstellung für diesen Parameters wird der Defaultmakroname **`standard`** verwendet.

Wie obiger Arbeitsschritt zeigt, stehen zum Platzieren von Labels dieselben Funktionen wie für normale Stromlaufsymbole zur Verfügung, d.h. auch hier besteht z.B. die Möglichkeit, Labels zu drehen, einen Standardwinkel für die Platzierung vorzugeben, usw.

Laden und platzieren Sie nun weitere Labels, so dass sich das in Abbildung 2-6 gezeigte Stromlaufbild ergibt (`Vss` an `V1.E`, `C101.1` und `S1006.2`; `Vdd` an `K10.A1` und `KK100.C`; `NET` an `IC10.B`). Bedienen Sie sich dabei nach Möglichkeit der Funktion Folge Symbol/Label zum wiederholten Platzieren desselben Labels. Testen Sie auch die Label- bzw. Signalnamensauswahl über Popupmenü. Dieses Netznamens-Popupmenü wird durch Betätigen einer beliebigen Maustaste oder durch die Eingabe eines Fragezeichens `?` auf die Abfrage nach dem Netznamen aktiviert. Im Netznamens-Popupmenü werden alle diejenigen Labels zur Auswahl angeboten, die bereits Anschlüsse in die aktuelle Netzliste besitzen, an die also eine Verbindung gelegt wurde. Der Button Alt im Netznamens-Popupmenü aktiviert die Funktion Folge Label.

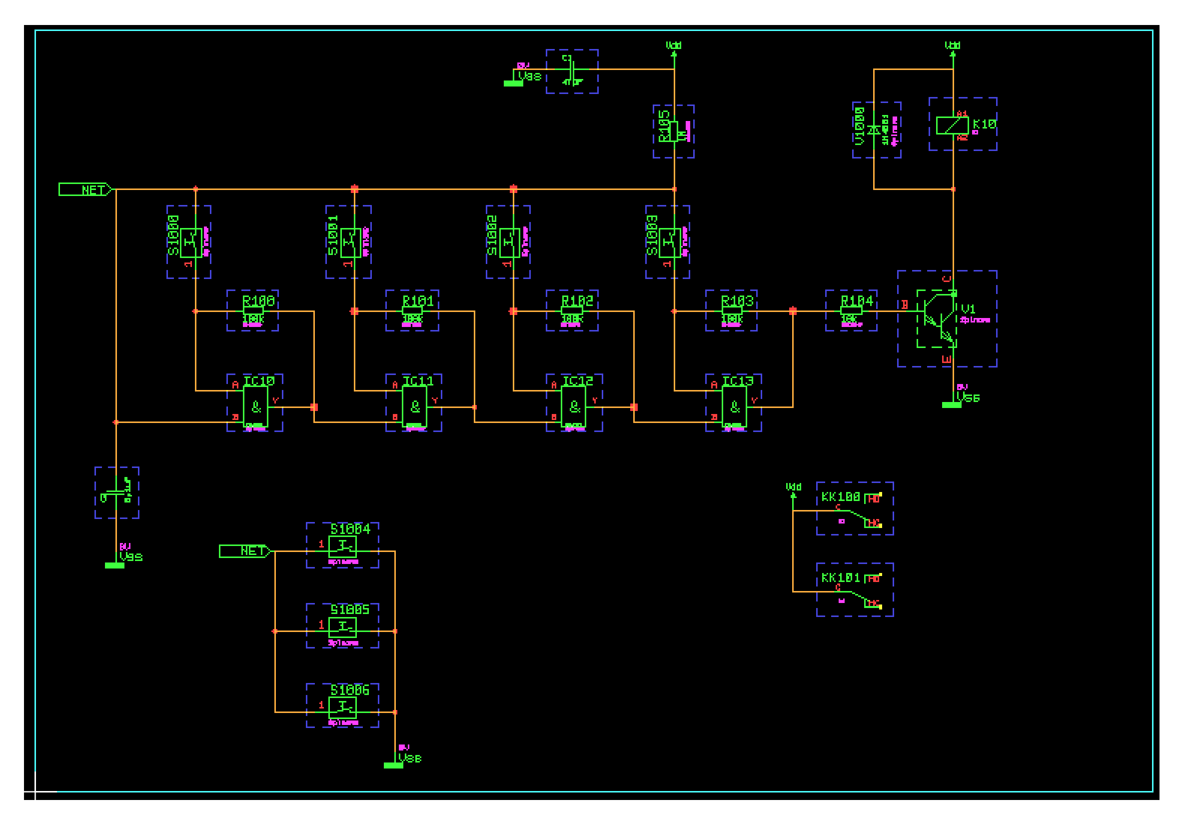


Abbildung 2-6: Stromlauf mit Symbolen, Verbindungen, Labels

Die Netzliste wird vom System im Hintergrund ständig aktualisiert. D.h., die Netzliste stimmt zu jedem Zeitpunkt der Bearbeitung exakt mit dem Schaltbild überein. Sie können dies mit folgenden Kommandos jederzeit überprüfen:



Die Funktion Highlight Netz im Menü Ansicht bewirkt eine grafische Kennzeichnung aller zum selektierten Netz gehörenden Verbindungen durch eine spezielle Farbe ("Highlight"). Dieses Highlight kann durch abermaliges Selektieren des Netzes über die Funktion Highlight Netz wieder zurückgesetzt werden. In **BAE HighEnd** bewirkt die Funktion Highlight Netz ein Highlight bzw. eine Highlight-Rücknahme der selektierten Netze in *allen* aktuell geladenen Plänen der aktuellen Projektdatei auf Schaltplan- und Layoutebene (globales Netz-Highlight, Cross-Probing).

Signalrouting

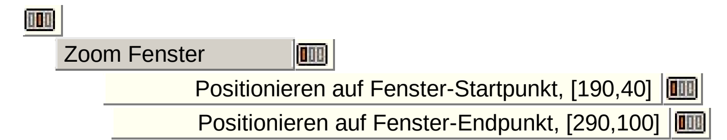
Im SCM ist ein Automatismus zum Umverlegen von Verbindungen beim Verschieben von Symbolen oder Labels und Gruppen auf Schaltplanebene integriert (Signalrouting). Es besteht die Möglichkeit, das Signalrouting wahlweise an- bzw. abzuschalten. Die Einstellung des Signalroutingmodus erfolgt über den Dialog Einstellungen aus dem Menü Einstellungen bzw. über die Optionen Signalrouting aus bzw. Signalrouting ein in dem über die rechte Maustaste aktivierbaren Untermenü der Funktion Bewegen Symbol/Label. Der dabei gewaehlte Signalroutingmodus wird als Bearbeitungsparameter mit dem aktuell bearbeiteten Schaltplan abgespeichert (siehe hierzu auch Kapitel 2.1.5).

Warnung

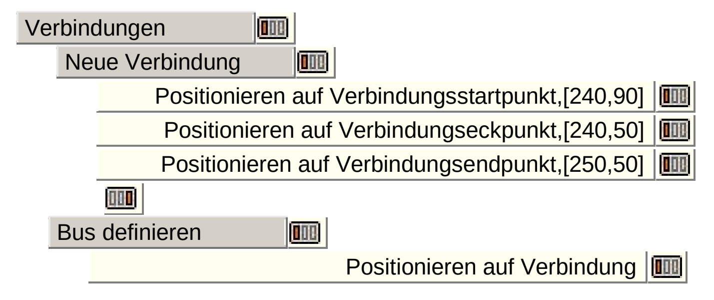
Beachten Sie, dass das im SCM integrierte Signalrouting nicht als optimiertes SCM-Autorouting konzipiert ist, sondern dem Anwender die Arbeit erleichtern soll. Dies erscheint uns zumindest für den Augenblick praktikabler als eine akademisch optimierte Lösung, auf die Sie u.U. sehr lange Zeit warten müssten! Wir sind uns dessen bewu&szligt, dass Sie das System an dieser Stelle relativ leicht "austricksen" können, also z.B. durch das Umplatzieren von Symbolen Netzlistenkonflikte herbeiführen können, auf die das System aufgrund von Mehrdeutigkeiten z.B. mit der Elimination von Verbindungen reagieren muss. Um konfliktbehaftete Umplatzierungen zu vermeiden, sollten Sie Symbole in (mehreren) hinreichend kleinen Schritten bewegen. Ist das Ergebnis des Signalrouters nicht zufriedenstellend, so steht immer noch die Undo-Funktion zur Zurücknahme der Änderung zur Verfügung.

Busse

Wir wollen nun die rechtsseitigen Anschlüsse der Relais-Kontakte **KK100** und **KK101** an einen Bus anschließen. Zoomen Sie hierzu zunächst in den geeigneten Arbeitsbereich:

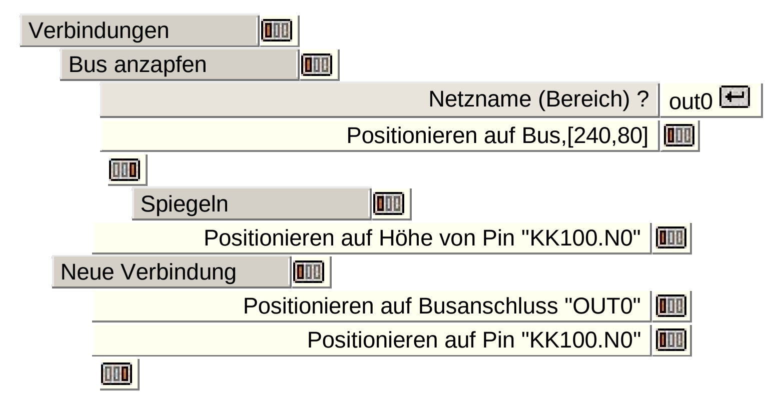


Definieren Sie nun mit den folgenden Kommandos einen Bus:



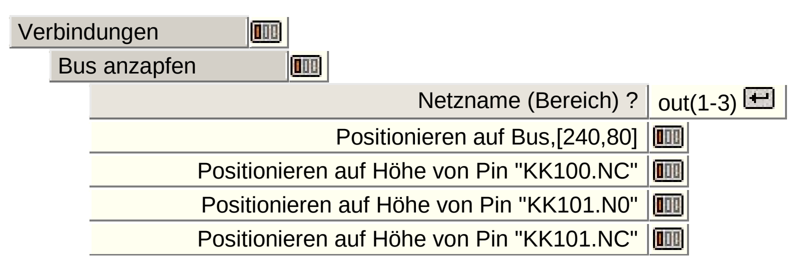
Zur Definition eines Busses wird zunächst eine anschlussfreie Verbindung gezeichnet. Das Selektieren dieser Verbindung über die Funktion Bus definieren bewirkt die Umwandlung der Verbindung in eine Busverbindung, die durch eine breitere Darstellung grafisch gekennzeichnet wird.

Definieren Sie nun mit folgenden Kommandos den Busanschluss **OUT0**, platzieren Sie diesen so, dass er mit einem Verbindungssegment an den Pin **N0** des Bauteils **KK100** angeschlossen werden kann, und verbinden Sie den Pin **N0** mit dem Busanschluss:



Nach der Abfrage des Netznamens für den Busanschluss lädt das System das Labelsymbol **bustap**, das anschließend auf dem Bus platziert werden kann, wobei das über die rechte Maustaste erreichbare Untermenü das Spiegeln des Busanschlusses auf die gegenüberliegende Seite des Busses ermöglicht. **bustap** ist ein spezielles Labelsymbol, das vom System zur Darstellung von Busanschlüssen verwendet wird (achten Sie darauf, dass dieses Symbol in der voreingestellten Bibliothek verfügbar ist).

Die Abfrage nach dem Netznamen erlaubt auch die Angabe von Netznamensbereichen. Platzieren Sie mit den folgenden Kommandos drei weiter Busanschlüsse mit den Netznamen **OUT1**, **OUT2** und **OUT3** (die zuvor eingestellte Spiegelfunktion bleibt aktiv bis zum nächsten Aufruf dieser Funktion):



Schließen Sie nun noch **KK100.NC** an **OUT1**, **KK101.N0** an **OUT2**, sowie **KK101.NC** an **OUT3** an.

Beachten Sie bitte auch die im Menü Verbindungen angebotenen Spezialfunktionen zum Bearbeiten bestehender Busse bzw. Busanschlüsse (Bustap bewegen, Bustap löschen, Bustap umbenennen).

Um Bussignale über mehrere Stromlaufblätter zu führen, besteht natürlich die Möglichkeit, Busse über Labels zu benennen. Laden Sie einen Label mit dem Netznamen **BUS**, und schließen Sie diesen Label an den soeben definierten Bus an (achten Sie dabei wieder darauf, dass der Kontaktbereich am Label erlischt):

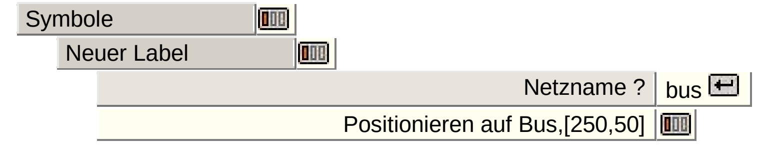


Abbildung 2-7 zeigt Beispiele für mögliche Busdefinitionen In Abbildung 2-7a sind die beiden Bussignale **S1** nicht miteinander verbunden, da sie auf verschiedenen Bussen liegen. In Abbildung 2-7b liegen die Bussignale **S1** auf demselben Bus und sind somit auch verbunden. Abbildung 2-7c zeigt die Definition von Sub-Bussen (**DATA** und **ADDR**); die Bussignale **S1** sind hierbei nicht verbunden, da sie auf unterschiedlichen Sub-Bussen liegen. Die Abbildung 2-7d und 2-7e zeigen die Verwendung von Labels zur Benennung von Bussen. Die Signale **S1** in Abbildung 2-7d liegen auf Bussen mit demselben Label und sind somit verbunden, während sie in Abbildung 2-7e auf Bussen mit verschiedenen Labels liegen und daher nicht verbunden sind.

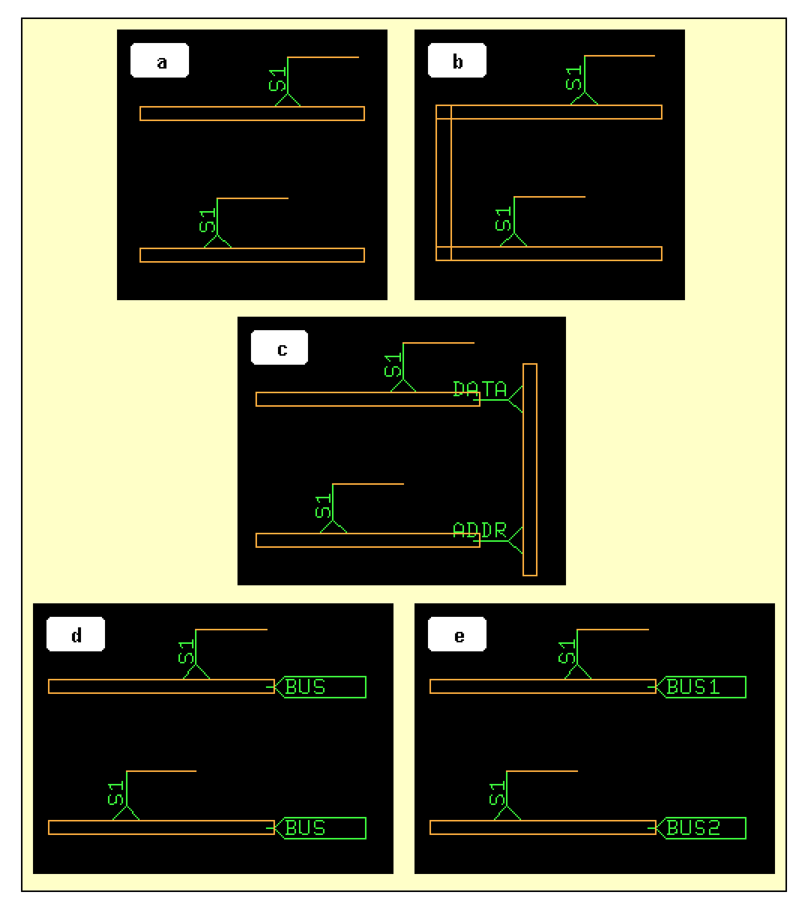
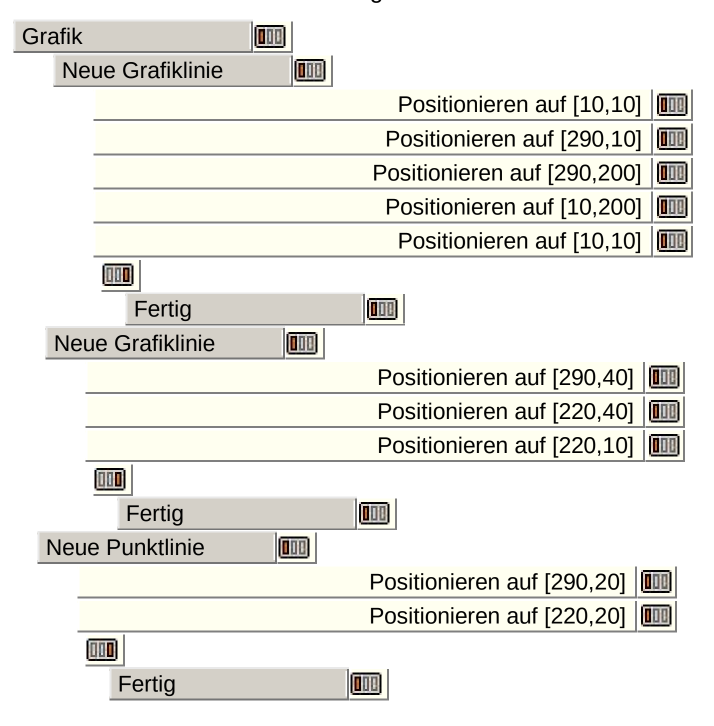


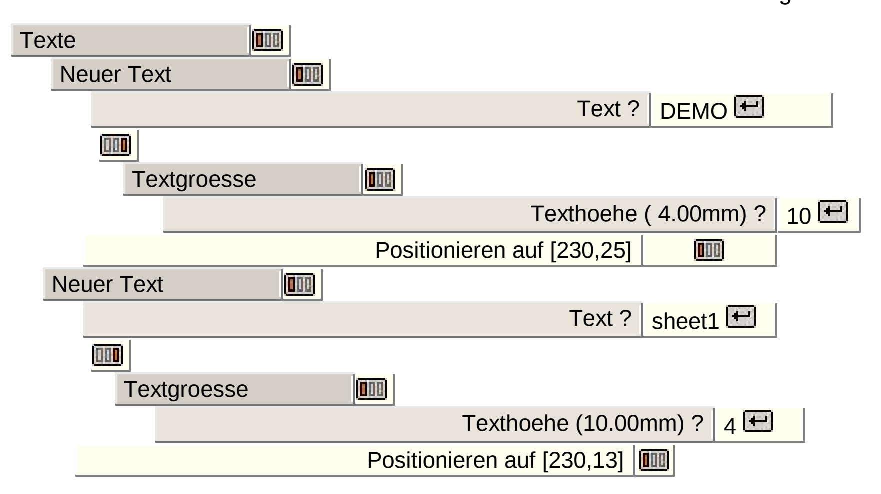
Abbildung 2-7: Busse im Bartels AutoEngineer

2.3.4 Text und Grafik

Selbstverständlich können auf Planebene auch Texte und Grafiken definiert werden. Mit folgenden Kommandos können Sie einen Grafikrahmen um ihr Schaltbild legen und ein Schriftfeld rechts unten im Schaltplan einzeichnen:



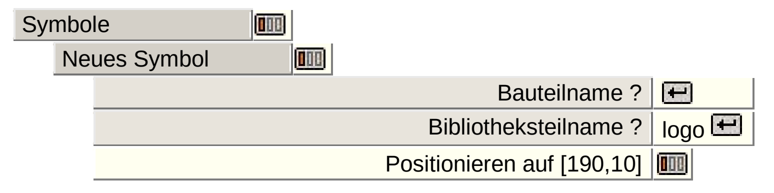
Mit folgenden Kommandos können Sie die Texte **`DEMO`** und **`sheet1`** in das zuvor gezeichnete Schriftfeld eintragen:



2.4 Spezielle SCM-Funktionen

2.4.1 Virtuelle Symbole

Schriftfelder, Firmenlogos, usw. lassen sich natürlich auch in Form von virtuellen Symbolen in der Bibliothek definieren und dann im Schaltbild platzieren. Mit folgenden Kommandos können Sie das Symbol `logo` mit dem Bartels Firmenlogo laden:



2.4.2 Gruppen

Ein mächtiges Werkzeug stellen die Gruppenfunktionen im **Schaltplaneditor** dar. Mit Hilfe dieser Funktionen können Teile des aktuell geladenen Schalplans oder Symbols zu Gruppen zusammengefasst und dann gespeichert, bewegt, gedreht, gespiegelt, kopiert oder gelöscht werden.

Die Gruppenfunktionen arbeiten nach dem Mengenprinzip. Es können Elemente wahlweise zur aktuell definierten Gruppe hinzugefügt (selektiert) oder auch wieder aus der Gruppe entfernt (deselektiert) werden. Die zur aktuell definierten Gruppe selektierten Elemente werden mittels Highlight angezeigt. Mit der Funktion Gruppe Polygon können mehrere in einem festzulegenden Polygonzug befindliche Objekte eines wählbaren Typs (Symbole/Labels, Verbindungen, Grafik, Texte, alle Typen) selektiert bzw. deselektiert werden. Die Funktion Gruppe Einzelelemente dient der Selektion bzw. Deselektion von Einzelelementen (Symbole/Labels, Verbindungen, Grafik, Texte). Hierbei besteht die Möglichkeit der repetitiven Objektauswahl, d.h. die Funktion bleibt mit den eingestellten Funktionsparametern solange aktiviert, bis kein gültiges Pickelement mehr angewählt wurde. Dadurch entfällt die sonst lästige Neuaktivierung der Auswahlfunktion bei der Selektion mehrerer Einzelelemente eines gewünschten Typs. Mit der Funktion Gruppe ruecksetzen können *alle* Elemente der aktuell definierten Gruppe deselektiert werden.

Alle zur aktuell definierten Gruppe selektierten Elemente werden in die nachfolgenden Gruppenfunktionen zum Speichern (Gruppe speichern), Bewegen (Gruppe bewegen), Kopieren (Gruppe kopieren), Löschen (Gruppe loeschen) einbezogen.

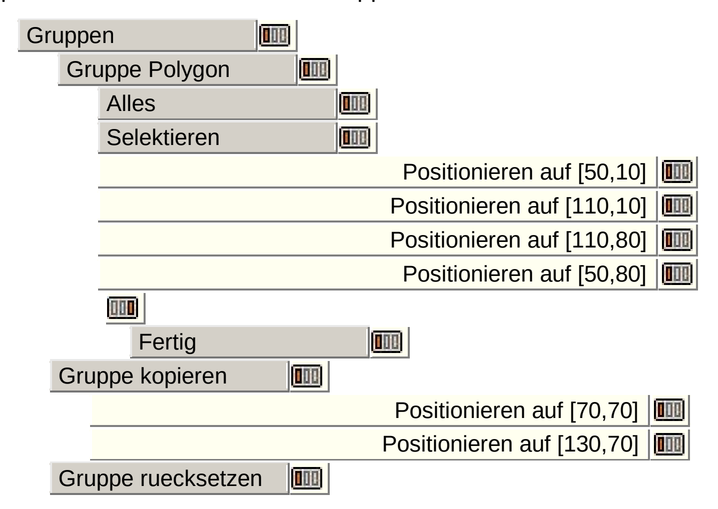
Mit der Funktion Gruppe speichern wird die aktuell definierte Gruppe als Dateielement abgespeichert. Hierbei ist ein Referenzpunkt zur Definition des Gruppenursprungs anzugeben. Das mit Gruppe speichern erzeugte Datenbankelement wird auf derselben Hierarchieebene angelegt wie das aktuell geladene Element. Um ein versehentliches Überschreiben existierender Datenbankelemente zu verhindern, aktiviert Gruppe speichern eine Bestätigungsabfrage für den Fall, dass ein Element mit dem spezifizierten Elementnamen bereits in der Zieldatei existiert. Mit Gruppe speichern lassen sich Teile eines erprobten Schaltplans oder Symbols in Form von Templates zur späteren Wiederverwendung abspeichern. Solche Templates (wie übrigens auch beliebige SCM-Pläne oder Symbole) können dann mit der Funktion Gruppe laden in andere Pläne bzw. Symbole geladen werden.

Während des Bewegens von Gruppen mit einer der Funktionen Gruppe bewegen, Gruppe kopieren oder Gruppe laden kann mit der rechten Maustaste ein Untermenü mit Funktionen zur Platzierung auf Relativ- oder Absolutkoordinaten (Sprung relativ, Sprung absolut) und zum Drehen bzw. Rotieren oder Spiegeln der Gruppe (Drehung links, Drehung rechts, Eingabe Winkel, Spiegelung aus, Spiegelung ein) aktiviert werden.

Mit der Funktion Gruppe Macroname können zur Gruppe selektierte Symbolmakros auf Schaltplanebene bzw. zur Gruppe selektierte Markermakros auf Symbolebene ersetzt werden. Diese Funktion eignet sich zum schnellen Austausch von Bauteil- oder Pinsymbolen (Technologieänderung). Die Attributzuweisungen und Textpositionen der ersetzten Originalelemente werden soweit möglich auf die neuen Elemente übertragen.

Die Funktion Gruppe laden setzt vor der eigentlichen Ladeoperation die aktuell definierte Gruppe zurück, d.h. es werden alle zum Zeitpunkt des Aufrufs der Funktion Gruppe laden selektierten Gruppenelemente deselektiert. Nachdem die Gruppe geladen ist, werden alle neu geladenen Gruppenelemente automatisch zur aktuellen Gruppe selektiert. D.h., die mit Gruppe laden geladenen Elemente (und nur diese) sind automatisch für die weitere Bearbeitung mit anderen Gruppenfunktionen selektiert.

Selektieren Sie mit den folgenden Kommandos die drei Schalter `S1004`, `S1005` und `S1006` mitsamt ihrer Beschaltung, und kopieren Sie diese so definierte Gruppe nach rechts:



Sie haben mit obigem Arbeitsschritt drei Symbole und zwei Labels platziert, sowie eine Reihe von Verbindungen erzeugt. Wie Sie sehen, hat das System auch die Bauteilbenennung für die kopierten Symbole automatisch weitergeführt. Die Labels wurden unverändert kopiert.

Wenn Sie alle Arbeitsschritte bis hierhin richtig ausgeführt haben, dann sollte das **`sheet1`** Ihres Schaltplans jetzt entsprechend Abbildung 2-8 aussehen.

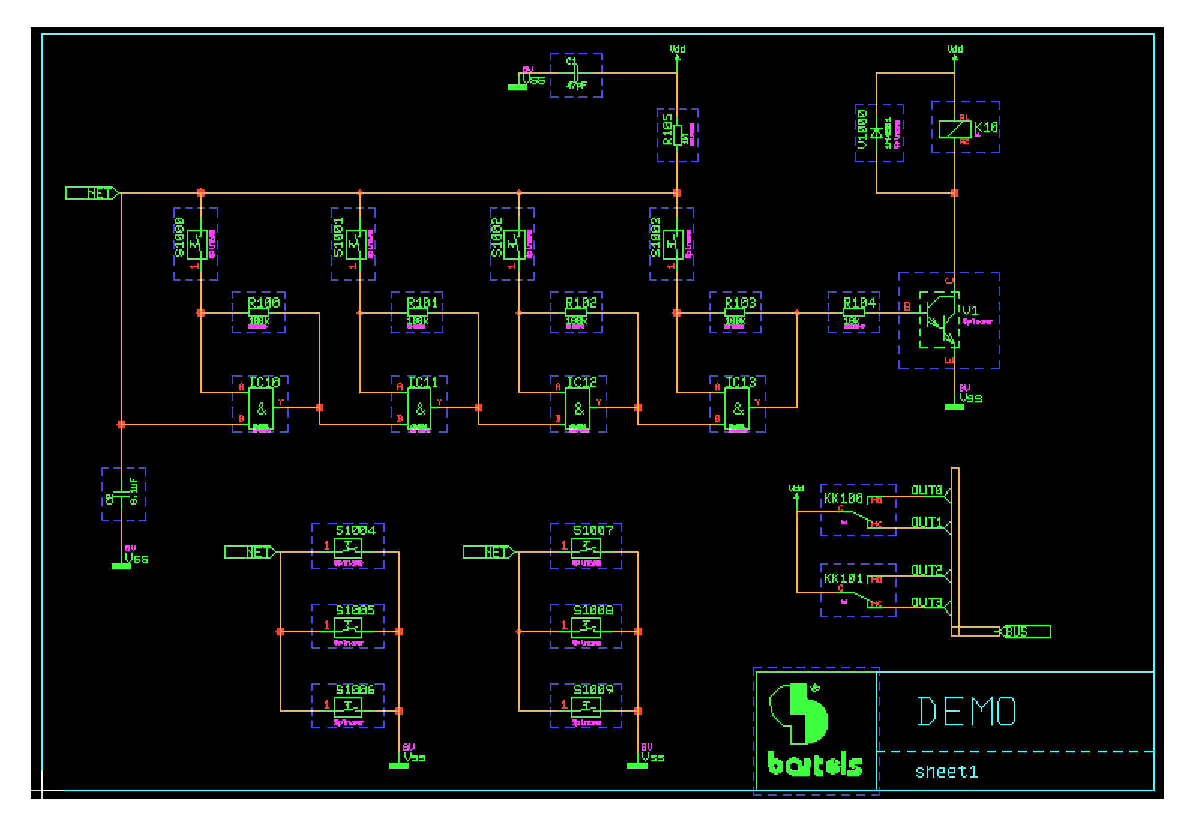
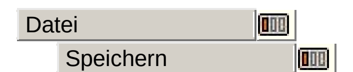


Abbildung 2-8: Stromlaufblatt Demo/Sheet1

Vergessen Sie in keinem Fall, Ihren *Stromlaufplan* mit folgenden Kommandos zu *sichern*:



Diesen Sicherungsvorgang können Sie übrigens jederzeit während der Bearbeitung durchführen. Sie sollten sich angewöhnen, grundsätzlich nach einer gewissen Bearbeitungszeit bzw. nach einer Reihe von Änderungen eine Sicherung durchzuführen, um einen etwaigen Datenverlust aufgrund z.B. eines Stromausfalls oder eines Rechnerabsturzes zu minimieren.

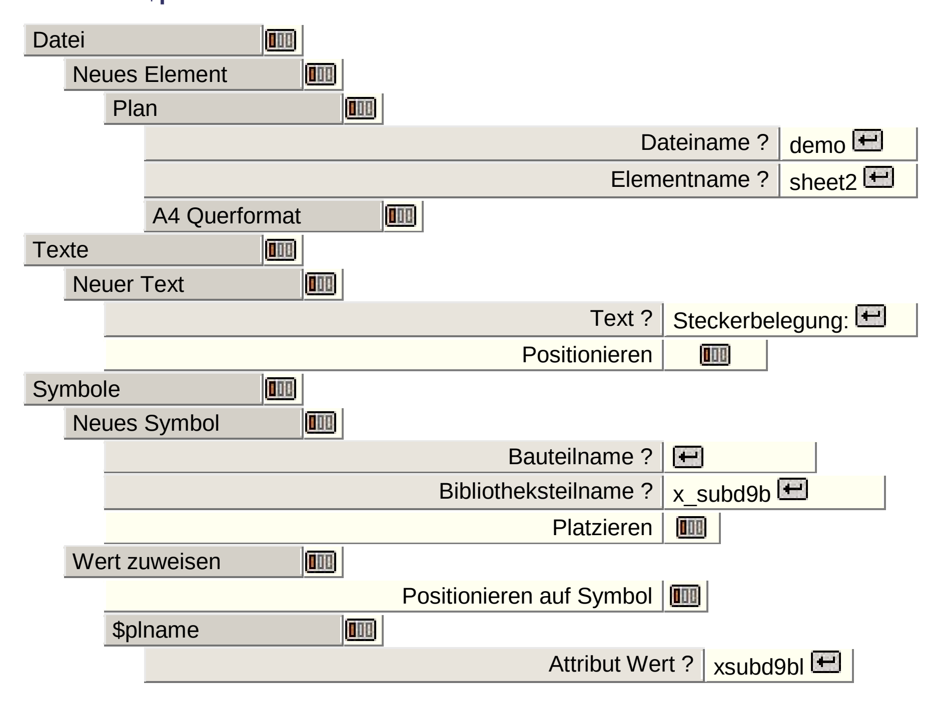
Gruppensymbolnummerierung

Das während der Platzierung von Gruppen über die rechte Maustaste erreichbare Untermenü der Funktionen Gruppe laden, Gruppe bewegen und Gruppe kopieren enthält die Option Neu benennen. Damit können alle in der Gruppen selektierten Symbole entsprechend dem Symbolnamensmuster neu benannt werden. Diese Funktion ist insbesondere nützlich, wenn aus einem Projekt mit hohen Bauteilnummern Teile in ein Projekt mit niedrigen Bauteilnummern übernommen werden sollen und man eine fortlaufende Bauteilnummerierung wünscht.

2.4.3 Steckerbelegung

Auf dem **sheet1** unseres Schaltplans haben wir die Schaltungslogik eingegeben. Was nun unter Umständen noch fehlt, ist z.B. die Steckerbelegung. Diese können wir auf einem zweiten Stromlaufblatt definieren.

Erzeugen Sie in der Projektdatei **demo.ddb** ein zweites Stromlaufblatt mit dem Namen **sheet2**, definieren Sie darauf den Text **Steckerbelegung:**, platzieren Sie das Steckersymbol **x_subd9b**, und geben Sie dem an diesem Symbol definierten Attribut **$plname** den Wert **xsubd9bl**:



Schließen Sie nun die Signale **Vss**, **Vdd** und die Signale **OUT(0-3)** des Busses **BUS** an das Steckersymbol an (Anschlussbelegung siehe Abbildung 2-9).

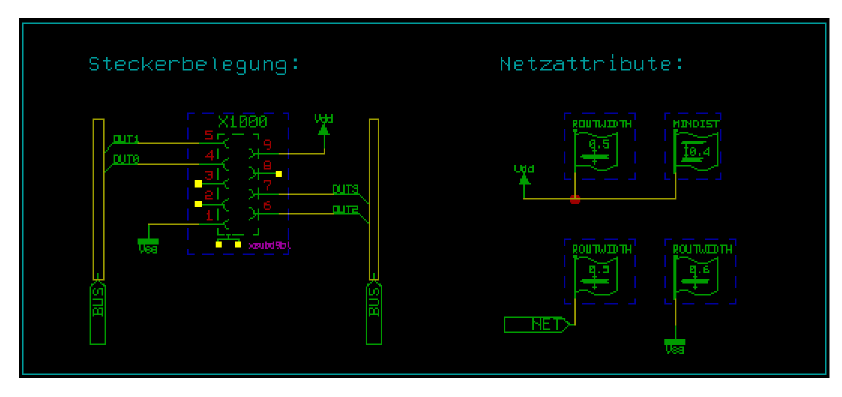


Abbildung 2-9: Stromlaufblatt Demo/Sheet2

2.4.4 Netzattribute

Im **Bartels AutoEngineer** besteht die Möglichkeit, Netzattribute zur Steuerung des **Autorouters** im Stromlaufplan zu definieren. So können z.B. netzbezogene Leiterbahnbreiten und Mindestabstände definiert werden.

Laden Sie den Label **NET** sowie das Symbol **att_rw** (Netzattribut **ROUTWIDTH**), verbinden Sie die Anschlüsse an **NET** und **att_rw** miteinander, und weisen Sie dem an **att_rw** definierten Attribut **$val** den Wert "0.5" zu, um den **Autorouter** zu veranlassen, das Signalnetz **NET** mit einer Breite von 0.5mm zu routen (siehe Abbildung 2-9).

Laden Sie nun den Label **Vdd**, und schließen Sie diesen an die Symbole **att_rw** (Netzattribut **ROUTWIDTH**; Werteintrag "0.5" für Attribut **$val**) und **att_md** (Netzattribut **MINDIST**; Werteintrag "0.4" für Attribut **$val**) an. Der **Autorouter** wird dieses Netz mit einer Breite von 0.5mm und einem Mindestabstand von 0.4mm zu potentialfremdem Kupfer routen.

Definieren Sie auf dieselbe Weise eine Leiterbahnbreite von 0.6mm für das Netz **Vss** (Laden Label **Vss** und verbinden zu Symbol **att_rw** mit Werteintrag "0.6" für Attribut **$val**).

Neben den oben beschriebenen Netzattribut-Definitionen können auch noch die Attribute **PRIORITY** und **POWWIDTH** zur Routersteuerung verwendet werden. Anmerkungen zur Wirkungsweise dieser Attribute finden sie in der Beschreibung für das Programm **LOGLIB**.

Tragen Sie schließlich noch den Text **Netzattribute:** in den Schaltplan ein, so dass sich das in Abbildung 2-9 gezeigte Schaltbild ergibt, und vergessen Sie nicht, Ihren *Schaltplan* zu *sichern*.

2.4.5 Tagsymbole

Der Symboltyp Tag dient der Zuweisung einzelner oder mehrerer Attributdefinitionen an Bauteile, Pins oder Netze bzw. an Gruppen derselben. Über Tags können darüber hinaus auch komplexe Informationen wie z.B. Vorgaben für Testabläufe oder logische Beziehungen zwischen Bauteilen, Pins, oder Netzen in das Design eingebracht werden.

Mit Hilfe der Funktion Symbol Tagmode aus dem Menü Symbole kann das aktuell geladene Symbol wahlweise in ein virtuelles Tag oder in ein Netzlistentag umgewandelt werden. Einträge und Zuweisungen über Netzlistentags werden vom **Packager** in die physikalische Netzliste transferiert und können somit im Layout weiterverarbeitet werden. Virtuelle Tags sind hingegen nur zur Bearbeitung auf Stromlaufebene vorgesehen.

Zur Darstellung von Tagsymbolen werden die Einträge Tag Symbol und Tag Link in der Farbauswahl angeboten. Bei der Ausgabe von Plots werden Tagsymbole üblicherweise nicht mitgeplottet.

Um die Zuweisung von Tagsymboleinträgen an Netze, Symbole oder Pins zu ermöglichen, müssen Pins mit entsprechendem Typ auf den Tagsymbolen platziert werden. Der Tagsymboltyp kann mit Hilfe der Option Tag Pin Funktion aus dem über die rechte Maustaste während der Pinplatzierung aktivierbaren Untermenü zugewiesen werden. Neben dem Modus Standard Pin (für Standardsymbole und Labels) stehen hierbei die Optionen Symbol Tag (zur Tagzuweisung an Symbole), Pin Tag (zur Tagzuweisung an Pins) und Netz Tag (zur Tagzuweisung an Netze) zur Auswahl.

Es sollte auch mindestens ein Attribut entweder durch Platzierung eines entsprechenden Attributnamens auf Tagsymbolebene oder mit fest vorgegebenem Attributwert in der Definition für die logische Bibliothek eingetragen werden. Bei der Definition des Eintrages für die logische Bibliothek ist zu beachten, dass alle auf dem Tagsymbol definierten Pins über das **LOGLIB**-Kommando **pin** eingetragen sein müssen. Reine Attributtage sind als **virtual** zu deklarieren.

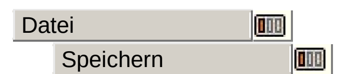
Auf Schaltplanebene können Tags wie normale Symbole mit der Funktion Neues Symbol platziert werden. Anschließend kann entweder implizit in der Platzierungsfunktion oder später explizit über die Funktion Symboltag zuweisen die Zuweisung von Tags vorgenommen werden. Hierbei können je nach Tagpintyp Symbole, Pins oder mit Label benannte Netze als Zielobjekte ausgewählt werden. Dabei kann für jeden Tagpin maximal ein Zielobjekt ausgewählt werden. Am Tagsymbol eingetragene Attributwerte werden dann automatisch an alle zugewiesenen Objekte übertragen. Die im BAE-System vordefinierten Spezialattribute (**$plname**, **$rpname**, **$routwidth**, **$powwidth**, etc.) behalten dabei ihre Bedeutung. Beachten Sie bitte, dass über **$routwidth** dabei auch die Möglichkeit der Spezifikation pinspezifischer Anschlussbreiten durch Eintrag des gewünschten Millimeterwertes besteht.

2.4.6 Templates

Beim Editieren von Schaltplänen werden Sie feststellen, dass Ihre Stromlaufpläne immer wiederkehrende Teile, wie etwa Schriftfeld-Informationen oder die oben beschriebenen Definitionen bezüglich Steckerbelegung bzw. Routersteuerung, enthalten. Das Datenbankkonzept des **AutoEngineer** ermöglicht die Speicherung derartiger Definitionen bzw. Schaltplanteile in Form von Templates (also wie Bibliothekselemente), die in neu zu erstellende Schaltplänen z.B. mit Hilfe der Funktionen zum Laden oder Speichern von Gruppen, der Funktion Ablegen auf Name aus dem Menü Datei, oder unter Verwendung des Utilityprogramms **COPYDDB** in neu zu erstellende Schaltpläne kopiert werden können.

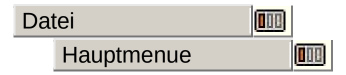
2.4.7 Verlassen des Stromlauf-Editors

Bevor Sie den Stromlauf-Editor verlassen, sollten Sie nicht vergessen, den gerade erstellten *Schaltplan* mit folgenden Kommandos zu *sichern*:

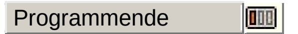


Rücksprung ins Hauptmenü

Um in die BAE-Shell zu gelangen, sind die folgenden Funktionen auszuführen:

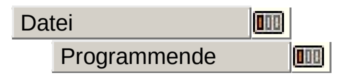


Sie befinden sich nun wieder in der Shell des BAE. Diese lässt sich wie folgt beenden:

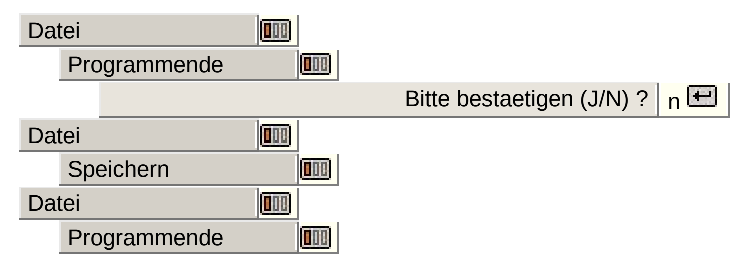


Programmende vom Schaltplanpaket aus

Der **Bartels AutoEngineer** kann auch direkt vom Schaltplanpaket aus beendet werden:

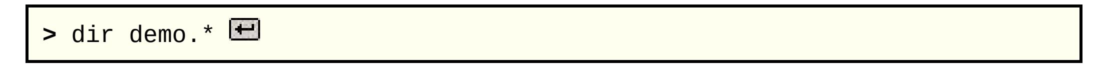


Es erfolgt der Rücksprung auf die Betriebssystemebene. Sollte das System mit einer Bitte um Bestätigung reagieren, dann wurde das aktuell geladene Element noch nicht gesichert. In diesem Fall sollten Sie die Funktion Programmende abbrechen, das im Speicher befindliche Element sichern und anschließend erst das Programmende herbeiführen:



Die nächsten Arbeitsschritte

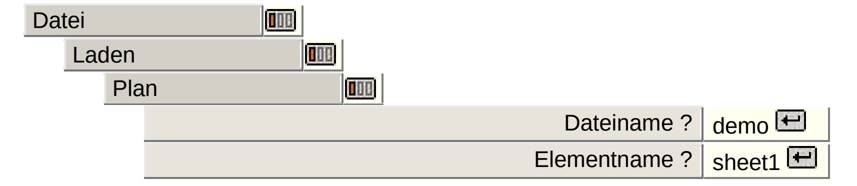
Auf Betriebssystemebene können Sie sich (z.B. unter DOS) mit folgendem Kommando die gerade generierte(n) Datei(en) auflisten lassen.



Die Projektdatei **`demo.ddb`** enthält nun den Stromlaufplan sowie eine logische, ungepackte Netzliste. Im nächsten Arbeitsschritt ist diese logische Netzliste mit Hilfe des **Packagers** (siehe Kapitel 3.2) in eine physikalische, gepackte Netzliste umzuwandeln. Anschließend erst kann mit dem Layoutsystem (siehe Kapitel 4) das Design der Leiterkarte erstellt werden.

2.5 SCM-Plotausgabe

Die SCM-Funktionen zur Erstellung von HP-GL-, HP-Laser- und Postscript-Ausgabedaten sind im Menü Plotausgabe des **Schematic Editors** zusammengefasst. Die Plotausgabe lässt sich nur starten, wenn ein Element geladen ist. Starten Sie also zunächst den **Schaltplaneditor** und Laden Sie mit den folgenden Kommandos das Stromlaufblatt `sheet1` aus der Datei `demo.ddb`:



2.5.1 Allgemeine Plotparameter

Ausgabekanal

Mit der Funktion Plotausgabe Kanal kann der Ausgabekanal angegeben werden, in den die Plotdaten gelenkt werden sollen. Dieser Kanal kann entweder direkt ein Ausgabegerät über einen Schnittstellennamen (z.B. `com2`, `lpt1` in DOS) oder den Namen einer Datei, in die die Ausgabedaten geschrieben werden, angeben. Dabei ist zu beachten, dass die Schnittstelle initialisiert ist bzw. genügend Platten-/Diskettenkapazität zur Aufnahme der Ausgabedaten bereitsteht. Sind diese Voraussetzungen nicht erfüllt, dann bricht die Ausgabe mit der Meldung `Schreiben ASCII-Datei fehlgeschlagen!` ab.

Wird kein Ausgabekanal explizit vorgegeben, dann erfolgt jeweils nach Aktivierung des entsprechenden Ausgabemenüpunktes eine Abfrage nach dem Namen des Ausgabekanals. In die Funktionen Plotausgabe Kanal, Plot HP-GL Ausgabe, Postscript Ausgabe und HP Laser Ausgabe im Menü Plotausgabe des **Schematic Editors** sind Popupmenüs zur schnellen Selektion der Plotdatei integriert. Aus Gründen der Datensicherheit werden hierbei Dateien mit den Endungen `.ass`, `.con`, `.ddb`, `.def`, `.exe`, `.fre`, `.ulc` und `.usf` ausgeblendet. Wahlweise besteht auch weiterhin die Möglichkeit, die Namen neu zu erstellender Plotdateien direkt über Tastatur einzugeben.

Maßstab

Mit der Funktion Massstab kann ein Vergrößerungsfaktor für die Plotausgabe angegeben werden. Der Defaultwert für den Maßstab beträgt 1.0. Es können Werte von 0.1 bis 100.0 spezifiziert werden.

Plot drehen

Über die Funktion Plot drehen kann gewählt werden, ob wie auf dem Bildschirm dargestellt geplottet werden soll (Drehung 0 Grad), oder ob die Ausgabe um 90 Grad gegen den Uhrzeigersinn gedreht erfolgen soll (Drehung 90 Grad). Dabei ist allerdings zu beachten, dass nicht alle Ausgabegeräte negative Koordinaten verarbeiten können. D.h., der Nullpunkt des auszuplottenden Elements ist vor der Plotausgabe ggf. mit der Funktion Nullpunkt im Menü Einstellungen entsprechend zu versetzen (z.B. in die linke obere Ecke des Plans).

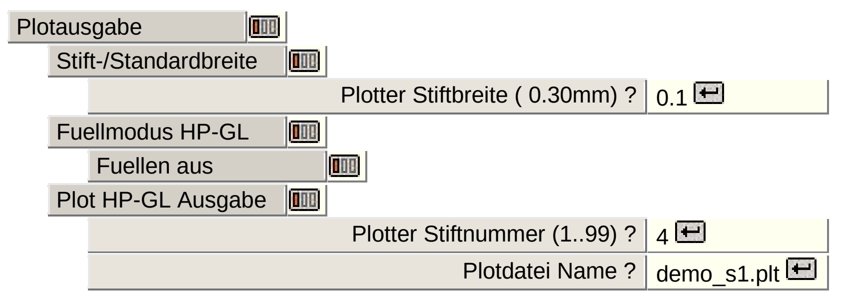
2.5.2 HP-GL Penplot

Die Ausgabe von HP-GL-Daten erfolgt mit einem Stift, dessen Breite im Menüpunkt Stift-/Standardbreite angegeben werden kann. Die Default-Stiftbreite beträgt 0.3mm; es können Werte von 0.01 bis 10.0 mm spezifiziert werden.

Die Plotgeschwindigkeit kann mit Geschwindigkeit beeinflusst werden. Dabei kann die Geschwindigkeit in cm/s oder **`S`** für maximale Geschwindigkeit angegeben werden. Die Defaulteinstellung ist **`S`**; als maximaler Zahlenwert wird 99 akzeptiert.

Für schnelle Kontrollplots kann im Menü Fuellmodus HP-GL mit Fuellen aus das Ausfüllen von Flächen unterbunden werden. Es wird dann beim Plotten von Flächen nur die Umrandungslinie gezeichnet. Mit Fuellen ein kann der Defaultzustand wieder eingestellt werden.

Die Plotausgabe wird mit dem Menüpunkt Plot HP-GL Ausgabe gestartet. Der Benutzer wird dabei zunächst um die Angabe der Stiftnummer (1..99) gebeten, und anschließend (sofern nicht explizit über Plotausgabe Kanal spezifiziert) nach dem Namen des Ausgabekanals gefragt. Um für das aktuell geladene Element einen HP-GL-Plot (Datei **`demo_s1.plt`**) ohne gefüllte Flächen mit dem Stift 4 (Breite 0.1mm) zu erzeugen, sind folgenden Kommandos auszuführen:

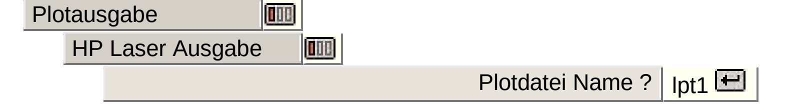


Das erfolgreiche Schreiben des HP-GL-Plots wird durch die Meldung **`HP-GL Plot beendet.`** quittiert.

Am Ende der HPGL-Ausgabedateien wird ein **`PG`**-Kommando zum Auswurf der Seite abgesetzt, um zu vermeiden, dass aufeinanderfolgende Blattausgaben übereinander geplottet werden.

2.5.3 HP-Laser-Ausgabe

Die Ausgabe von HP-Laser-Plots im Format PCL (Printer Command Language) kann mit dem Menüpunkt HP Laser Ausgabe gestartet werden. Der Plot wird dabei automatisch auf A4 skaliert, d.h. weder die Einstellung des Maßstabs, noch die Angabe einer Stift- bzw. Standardbreite haben hier Wirkung. Um den aktuell geladenen Schaltplan auf **`lpt1`** (d.h. hier z.B. Direktangabe der DOS-Schnittstelle zu Laserdrucker) auszugeben, sind folgende Kommandos auszuführen:

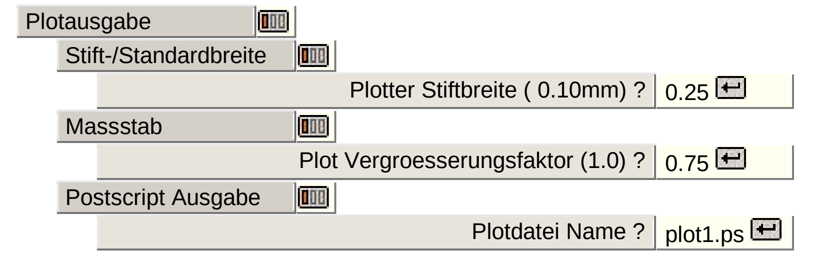


Das erfolgreiche Schreiben des PCL-Plots wird durch die Meldung **`HP Laser Ausgabe beendet (Skalierung 1:...).`** mit der Angabe des zur Skalierung auf Blattgröße verwendeten Skalierungsfaktors quittiert. Die Daten müssen im Binärmodus an den Ausgabekanal übertragen werden. Erfolgt die Ausgabe zunächst auf eine externe Datei (z.B. **`pclplot`**), dann ist auf DOS-Ebene bei der anschließenden Übertragung dieser Datei an den Laserdrucker mit Hilfe des COPY-Befehls die Option **`/b`** anzuwenden:

```
> copy pclplot lpt1 /b
```

2.5.4 Postscript-Ausgabe

Die Postscript-Ausgabe wird mit der Funktion Postscript Ausgabe gestartet. Um den aktuell geladenen Schaltplan mit einem Vergrößerungsfaktor von 0.75 und unter Verwendung einer Standardlinienbreite von 0.25 mm auf die Datei `plot1.ps` auszugeben, sind folgende Kommandos auszuführen:

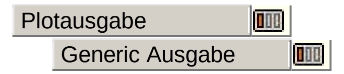


Das erfolgreiche Schreiben der Postscript-Datei wird durch die Meldung `Postscript Ausgabe beendet.` quittiert.

2.5.5 Generische Ausgabe unter Windows

In den Windows-Versionen der BAE-PC-Software ist eine generische Plot- bzw. Druckerausgabe implementiert. Damit werden durch den **Schaltplaneditor** der BAE-Windows-Software prinzipiell *alle* unter der aktuell definierten Windows-Betriebssystemkonfiguration verfügbaren Print- bzw. Plotfunktionen unterstützt.

Zur Aktivierung des Windows Print-/Plot-Menü sind die folgenden Kommandos auszuführen:



Die im Windows-Druckerdialog vorgenommenen Einstellungen für die Anzahl der Kopien, die Sortierung sowie den Seitenausgabebereich werden bei der generischen Ausgabe berücksichtigt.

Bei Anwahl der Option Alle Seiten im Druckerdialog der generischen Ausgabe unter Windows werden alle Seiten ausgegeben. Somit ist es möglich z.B. alle Stromlaufblätter eines Projektes auf einmal auszudrucken. Um z.B. Stromlaufblätter gemischt gedreht und nicht gedreht ausgeben zu können, werden jeweils die für das zu plottende Element eingestellten Druckparameter berücksichtigt. Diese können sich von den Parametern des aktuell geladenen Elements unterscheiden.

Bei Anwahl der Option Markierung im Druckerdialog der generischen Ausgabe unter Windows kann ein Bereich für die Plotausgabe selektiert werden.

Bei der generischen Ausgabe wird eine automatischen Anpassung der Skalierung auf das für die Druckausgabe definierte Blattformat unter Beibehaltung des Seitenverhältnisses vorgenommen, wenn die Größe des zu plottenden Elements die über die Druckereinstellung definierte Blattgröße überschreitet. Die Funktion Generic Ausgabe zeigt nach Beendigung der Ausgabe in der Mitteilungszeile den zur Skalierung auf Blattgröße verwendeten Skalierungsfaktor an.

2.5.6 Bitmap-Plotausgabe auf die Windows-Zwischenablage

In der Windowsversion ermöglicht Ausgabe nach Zwischenablage aus dem Menü Plotausgabe die Ausgabe von Zeichnungsdaten in eine Bitmap, die in die Zwischenablage zum weiteren Verarbeiten durch (Einfügen) in andere bitmapfähige Windowsanwendungen übertragen wird. Per Default wird das gesamte aktuell geladene Element ausgegeben. Mit Clipping ein lässt sich die Ausgabe auf ein mausselektierbares Rechteck beschränken. Die Dialogbox zur Plotparametereinstellung erlaubt auch eine Größenvorgabe für die Bitmap, sowie die Auswahl des Rotationsmodus für die Ausgabe.

2.6 Hierarchischer Schaltungsentwurf

Bei der Schaltplaneingabe besteht die Möglichkeit des hierarchischen Designs. D.h. es ist möglich, Schaltpläne als Blockschaltbilder zu definieren, und diese auf anderen Stromläufen als Block zu referenzieren. Wir empfehlen diese relativ arbeitsintensive Vorgehensweise allerdings nur in Fällen, wo aufgrund des Umfangs und der Struktur des Stromlaufplans auch wirklich eine hinreichend begründete Veranlassung dazu besteht. In der Regel ist dies z.B. beim Entwurf integrierter Schaltungen (Gate Arrays, Standardzellen, ASICs) der Fall.

2.6.1 Blockschaltbild

Für das hierarchische Design ist zunächst ein Blockschaltbild (auf einem oder mehreren Stromlaufblättern) zu zeichnen. Die Kennzeichnung eines Blockschaltbildes als Sub-Block erfolgt durch die Definition eines Blocknamens mit Hilfe der Option Sub-Block der Funktion Plan Blockname aus dem Dialog Einstellungen. Ist ein Blockschaltbild auf mehrere Blätter verteilt, so ist jeweils der gleiche Blockname einzutragen. Abbildung 2-10 zeigt ein Beispiel für ein Blockschaltbild (der Blockname ist dabei definiert als **BLOCK**). Die Anschlüsse des Blockschaltbildes zu den hierarchisch übergeordneten Stromlaufblättern erfolgt über Modulports. Modulports können mit der Funktion Neuer Modulport aus dem Menü Symbole geladen werden; das System verwendet per Konvention das Labelsymbol **port** für die Darstellung von Modulports. In der Abbildung 2-10 sind die Modulports **S**, **R**, **Q**, **/Q** sowie **HYPER** definiert. Die beim Hierarchischen Design notwendige Unterscheidung zwischen lokalen und globalen Netzen erfolgt mit Hilfe des Zeichens **&** bei der Definition von Netznamen. Durch Voranstellen dieses Zeichens vor einem Netznamen wie bei Netz **&ABC** bzw. Netz **/&ABC** in Abbildung 2-10 werden die entsprechenden Signale lokal in Bezug zum Sub-Block definiert. Die herkömmliche Netznamensvergabe (wie z.B. bei **VCC**) bewirkt die Referenzierung eines globalen Netzes. Bei der Vergabe von globalen Netznamen in Blockschaltbildern ist prinzipiell zu bedenken, dass diese mit allen gleichnamigen Netzen aller anderen Hierarchieebenen verbunden sind.

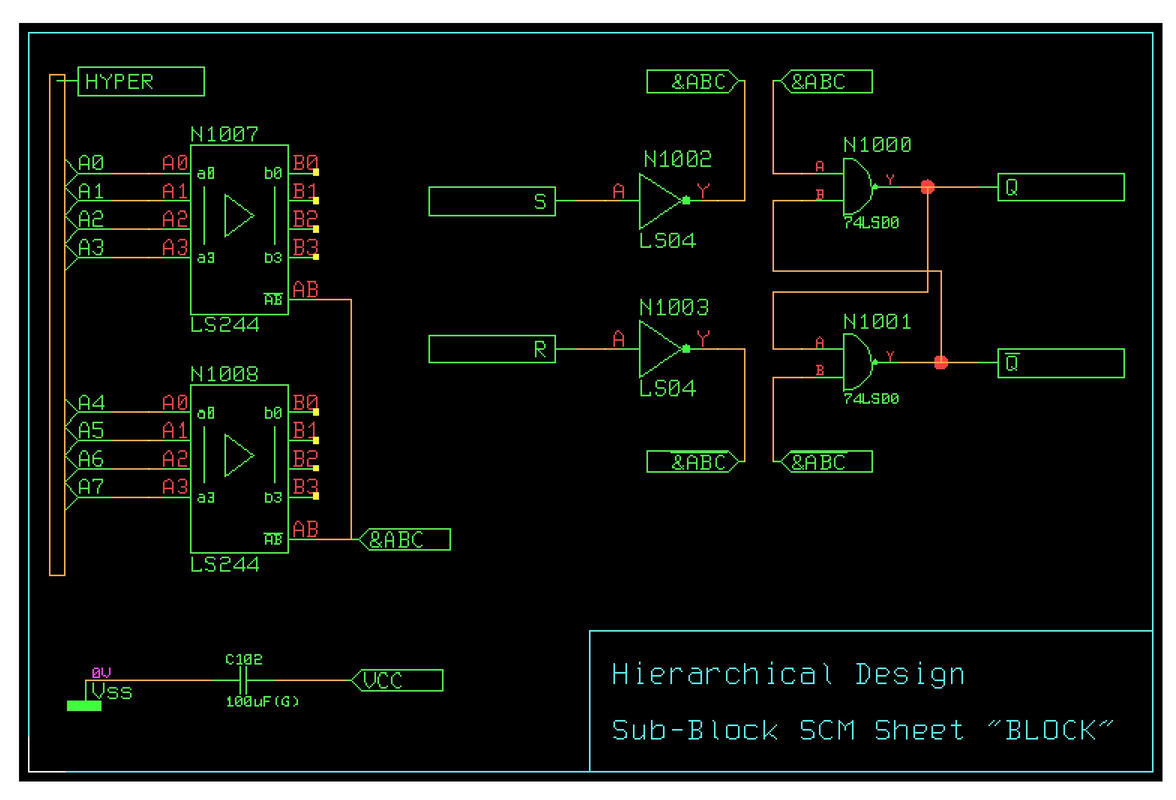


Abbildung 2-10: Hierarchischer Schaltungsentwurf; Blockschaltbild "BLOCK"

Zur besseren Unterstützung bei der Erstellung und Bearbeitung von Blockschaltbildern steht in der Funktion Plan Blockname zusätzlich die Option Einzel-Sub-Block zur Verfügung. Ein Einzel-Sub-Block darf im Gegensatz zum normalen Sub-Block nur einmal referenziert werden. Ein Einzel-Sub-Block wird vom **Packager** und der Backannotation wie ein normales Schaltplanblatt behandelt, d.h. die Symbolnamen werden ohne Zusatz eines **[Pn]**-Prefix in das Layout übernommen, und Bauteilnamensänderungen und Pin-/Gate-Swaps werden von der Backannotation in den Blockschaltplan zurückgemeldet.

2.6.2 Blocksymbol

Zur Referenzierung eines Blockschaltbildes in einem hierarchisch übergeordneten Stromlaufblatt ist die Definition eines speziellen Stromlaufsymbols notwendig, wobei die Liste der Pins dieses Symbols mit der Liste der Modulports im Blockschaltbild übereinstimmen muss.

Neben der Definition des Blocksymbols wird für den späteren **Packager**-Lauf zur Erzeugung der physikalischen Netzliste ein entsprechender Eintrag in der logischen Bibliothek benötigt (siehe hierzu auch die Kapitel 7.11 und Kapitel 3.2).

Abbildung 2-11 zeigt das Blocksymbol **DFF** für das Blockschaltbild aus Abbildung 2-10 sowie die zugehörige **LOGLIB**-Definition zur Referenzierung des Blockschaltbildes aus Abbildung 2-10.

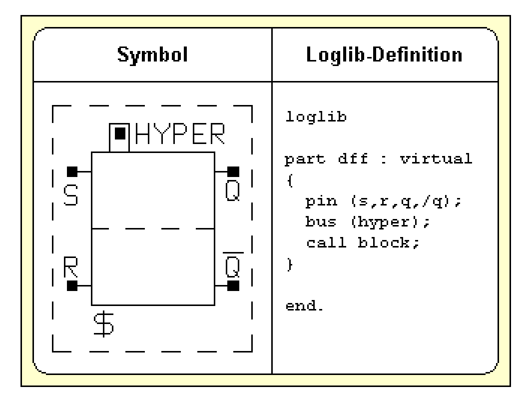


Abbildung 2-11: Hierarchischer Schaltungsentwurf; Blocksymbol "DFF" mit Loglib-Definition

Wie aus dem Beispiel zu ersehen ist, ist das Blocksymbol als Virtuelles Bauteil zu definieren; die Referenzierung des Blockschaltbildes erfolgt mit Hilfe des Befehls `call`. Beachten Sie bitte auch, dass in unserem Beispiel die Definition eines Busses am Blocksymbol verwendet wurde.

2.6.3 Top-Level-Schaltbild

Abbildung 2-12 zeigt die Verwendung des in Abbildung 2-11 dargestellten Blocksymbols **DFF** in einem hierarchisch übergeordneten Stromlauf; das Blockschaltbild **BLOCK** wird dabei dreimal (**DFF_1**, **DFF_2** und **DFF_3**) über das Symbol **DFF** referenziert.

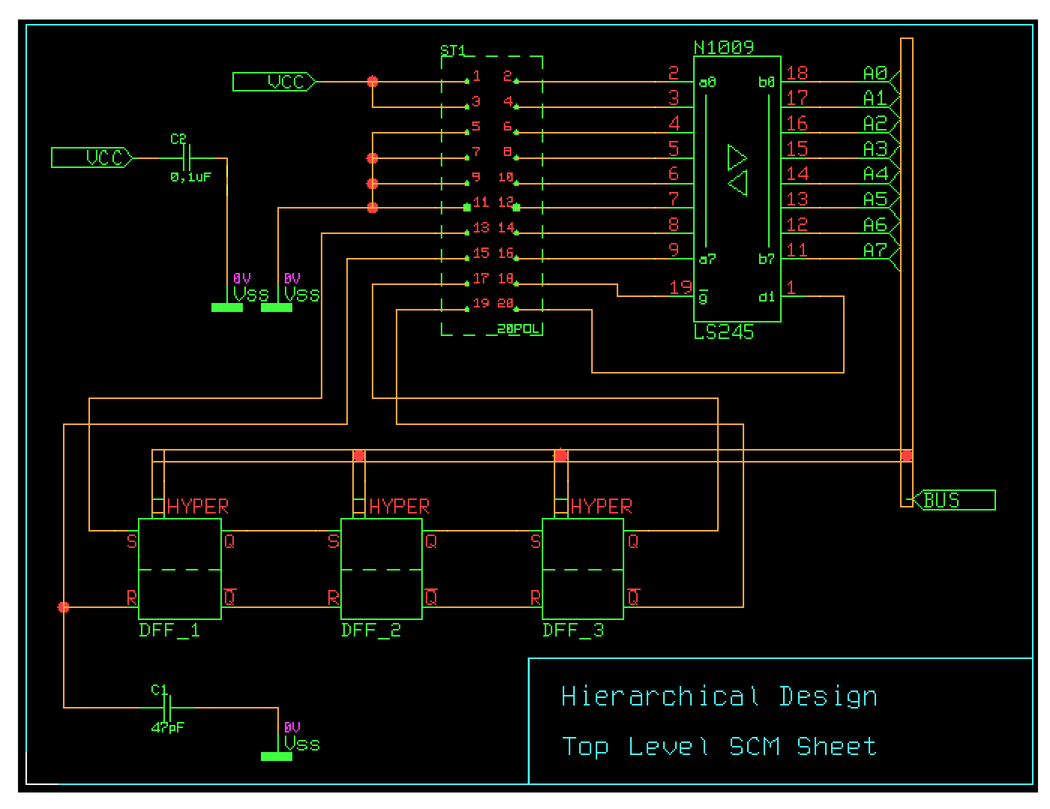


Abbildung 2-12: Hierarchischer Schaltungsentwurf; Top-Level-Schaltbild

2.7 Backannotation

Backannotation ist immer dann durchzuführen, wenn im Layout Änderungen an der Netzliste (Umbenennung von Bauteilen, Pin/Gate Swaps oder Zuweisung alternativer Gehäusebauformen) vorgenommen wurden, die in den Stromlauf zurückgemeldet werden müssen. Bauteile können mit der Funktion Name in Netzliste im Menü zur manuellen Bauteilplatzierung umbenannt werden. Pin/Gattertausch kann entweder interaktiv mit der Funktion Pin/Gate Swap im Menü zur manuellen Bauteilplatzierung oder automatisch mit den Funktionen Voll-Autoplace, Einzeloptimierung bzw. Multi-Optimierung **Autoplacement** durchgeführt werden. Darüber hinaus kann auch die Option Router P/G-Swap Ein im **Autorouter** die automatische Anwendung von Pin/Gate Swaps während des Rip-Up-Routings bewirken. Die Zuweisung alternativer Gehäusebauformen erfolgt mit der Untermenüfunktion Bauform aendern während der manuellen Bauteilplatzierung im Layout. Es ist dringend zu beachten, dass alle oben genannten Modifikationen in der physikalischen Netzliste verloren gehen, wenn sie vor einer neuerlichen Schaltplanbearbeitung nicht mit Backannotation in den Stromlauf zurückgemeldet werden.

Die Backannotation ist vollständig im **Schaltplaneditor** integriert und kann über das Kommando Backannotation aus dem Menü Diverse bzw. Utilities gestartet werden. Außerdem sind die SCM-Funktionen zum Laden von Elementen sind mit einem Mechanismus zur wahlweisen automatischen Durchführung notwendiger Backannotation-Prozesse beim Laden von Stromlaufplänen in den **Schaltplaneditor** ausgestattet. Hierzu werden layoutspezifische Datenbankeinträge ausgewertet, die im Layoutsystem beim Speichern generiert werden, wenn Pin-/Gate-Swaps oder Bauteilnamensänderungen vorgenommen wurden. Ist beim Laden eines SCM-Plans eine Anforderung zur Durchführung der Backannotation vorhanden, dann wird automatisch eine Bestätigungsabfrage aktiviert, die wahlweise die Durchführung der Backannotation ermöglicht. Nach dem Backannotation-Lauf wird der Datenbankeintrag für die Backannotation-Anforderung wieder gelöscht.

Die Anwendung der Backannotation ist im Kapitel 3 näher erläutert.

Kapitel 3 Packager / Backannotation

Dieses Kapitel beschreibt die Arbeitsweise von **Packager** und **Backannotation**. Anhand von Beispielen wird die Handhabung des **Packagers** zum Zwecke der Umwandlung logischer in physikalische Netzlisten, d.h. die Forward-Annotation vom Stromlaufpaket zum Layout erklärt. Weiterhin wird die Vorgehensweise bei der Rückmeldung von im Layout vorgenommenen Netzlistenänderungen in den Stromlauf mit Hilfe der Backannotation aufgezeigt. Hierbei wird der Schaltplanentwurf aus dem vorherigen Kapitel für eine weitere Bearbeitung in den nachfolgenden Kapiteln aufbereitet. Es wird daher empfohlen, dieses Kapitel Schritt für Schritt und ohne Auslassung irgendwelcher Abschnitte durchzuarbeiten, um einen vollständigen Überblick über die Arbeitsweise von **Packager** und Backannotation zu gewinnen. Sobald ein spezielles Kommando angewandt bzw. dessen Benutzung erläutert wurde, ist davon auszugehen, dass der Leser dieses Kommando verstanden hat und es bei Bedarf ohne nähere Erläuterungen wieder ausführen kann. Nachfolgende Instruktionen zu dem betreffenden Kommando sind dann weniger ausführlich, um das Lesen zu vereinfachen und den Lernprozess zu beschleunigen.

Inhalt

Abbildungen

3.1 Allgemeine Hinweise

Das Schaltplanpaket des **Bartels AutoEngineer** besteht im Wesentlichen aus einem grafisch-interaktiven **Schaltplaneditor** mit integrierter Backannotation sowie dem Programm-Modul **Packager** zur Umwandlung von logischen in physikalische Netzlisten ("Forward Annotation"). Die nachfolgenden Abschnitte dieses Benutzerhandbuchs enthalten eine detaillierte Beschreibung des **Packager**-Moduls und der Backannotation.

3.1.1 Komponenten und Leistungsmerkmale

Die Umsetzung von Stromläufen in das Layout, d.h. die Übergabe der Netzliste geschieht im **Bartels AutoEngineer** mit Hilfe des **Packager** und umgekehrt, also vom Layout in den Stromlauf (nach etwaigen Modifikationen der Netzliste) mit **Backannotation**.

Der Designfluss hinsichtlich der Netzlistenverarbeitung funktioniert im **Bartels AutoEngineer** mit Hilfe von **Packager** und **Backannotation** nach dem in Abbildung 3-1 dargestellten Schema.

Nach Erstellung bzw. Änderung des Schaltplans mit Hilfe des **Schematic Editors** ist die dadurch generierte logische Netzliste mit Hilfe des **Packagers** in das Layout zu transformieren. Der **Packager** hat dabei die Aufgabe, entsprechend der in der Bibliothek definierten Zuordnung der Stromlauf- zu den Layoutsymbolen (siehe hierzu auch die Beschreibung des Programms **LOGLIB**), die im Schaltplan definierten Stromlaufsymbole in die entsprechenden Layoutgehäuse zu packen und die daraus resultierende Pinzuordnung zu definieren. Ergebnis des **Packager**-Laufs ist eine gepackte physikalische Netzliste.

Werden im Layout Modifikationen an der Netzliste (durch Änderung von Bauteilnamen, Pin/Gate Swaps, Zuweisung alternativer Gehäusebauformen, usw.) vorgenommen, dann ist die modifizierte physikalische Netzliste mit Hilfe der im **Schaltplaneditor** integrierten **Backannotation**-Funktion wieder in den Stromlauf, d.h. in eine modifizierte logische Netzliste zurückzutransferieren.

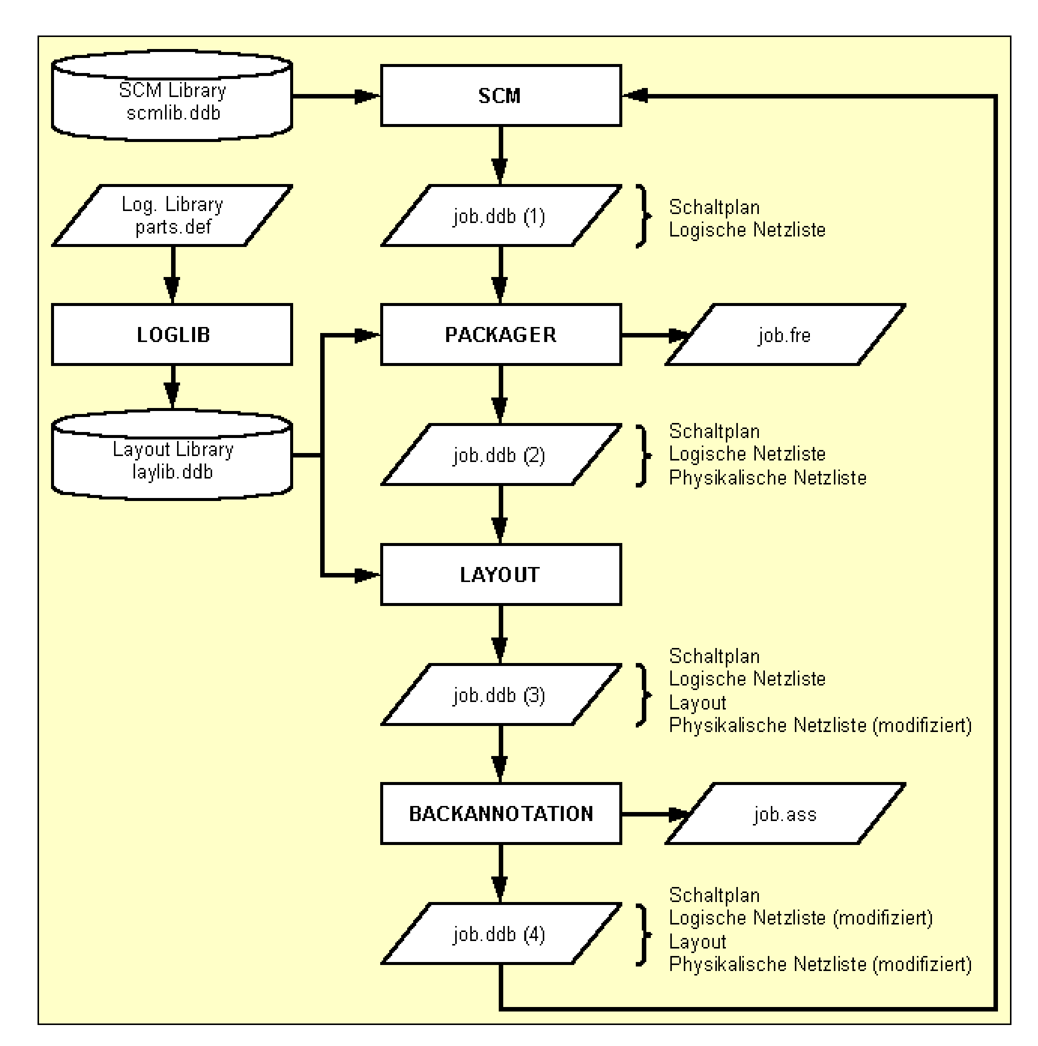


Abbildung 3-1: Designfluss Packager - Backannotation

3.2 Packager

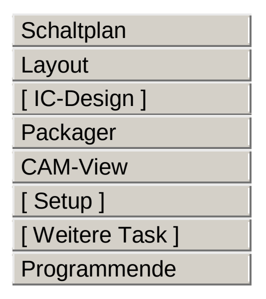
Ein **Packager**-Lauf ist immer dann durchzuführen, wenn im Stromlauf Änderungen an der Netzliste (Laden neuer Symbole, Umverlegen von Verbindungen, Änderung von Attributwerten, usw.) vorgenommen wurden, die in das Layout gemeldet werden müssen.

3.2.1 Programmaufruf

Der Aufruf des **Packagers** erfolgt aus der Shell des **Bartels AutoEngineer** und sollte grundsätzlich in dem Verzeichnis vorgenommen werden, in dem die zu bearbeitende Projektdatei abgelegt ist. Die BAE-Shell wird von Betriebssystemebene aus mit folgendem Befehl gestartet:



Der **AutoEngineer** zeigt auf dem Schirm das Bartels-Logo sowie folgendes Menü (die Funktion Setup ist nur unter Windows bzw. Motif verfügbar; die Menüpunkte IC-Design und Weitere Task sind nur in speziellen Softwarekonfigurationen wie etwa in **BAE HighEnd** oder **BAE IC Design** verfügbar):



Über das Kommando



wird der **Packager** des **AutoEngineer** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

3.2.2 Hauptmenü

Nach dem Aufruf des **Packagers** befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden des **Packagers** ist die Funktion Parameter mit dem zugehörigen Menü aktiviert. Das Hauptmenü ist ständig verfügbar und enthält folgende Funktionen:

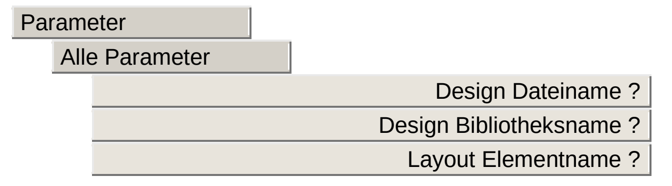


Mit der Funktion Parameter werden die Parameter für den mit Start zu startenden **Packager**-Lauf eingestellt. Mit den Funktionen Hauptmenue, Schaltplan und Layout kann in das BAE-Hauptmenü, in den **Schaltplaneditor** oder in das Layoutsystem gewechselt werden. Programmende beendet die BAE-Sitzung. Beim Wechsel in das Layoutsystem wird automatisch die Erzeugung eines Layoutelements für die zuletzt vom **Packager** erzeugte Netzliste vorgeschlagen, wenn noch kein entsprechendes Layoutelement existiert.

3.2.3 Programmablauf

Vor der Durchführung des eigentlichen **Packager**-Laufs mit der Funktion Start sind über die entsprechenden Funktionen aus dem Menü Parameter die Parameter für den **Packager**-Lauf einzustellen.

Nach dem Aufruf der Funktion Alle Parameter hat der Benutzer nacheinander folgende Abfragen zu beantworten:



Diese Parameter können auch einzeln über die entsprechenden Funktionen aus dem Menü Parameter spezifiziert werden.

Für den Design Dateinamen ist der Name der Projektdatei einzugeben, für die die Netzlistenumsetzung durchgeführt werden soll. Die Projektdatei muss mit der Extension `.ddb` verfügbar sein; der Dateiname ist ohne diese Extension anzugeben. Wird ein Leerstring (Betätigen der Eingabetaste ↵) für den Dateinamen eingegeben, dann verwendet das System den Dateinamen des im vorhergehenden Programm-Modul des **Bartels AutoEngineer** geladenen Elements, d.h. den systemweiten Projektnamen.

Auf die Abfrage nach dem Design Bibliotheksnamen ist der Name der Layoutbibliothek einzugeben, die für das Leiterkartendesign verwendet werden soll, und in der auch die für die Netzlistenumsetzung notwendigen Informationen der logischen Bibliothek enthalten sind. Die Bibliotheksdatei muss mit der Extension `.ddb` verfügbar sein, der Dateiname ist ohne diese Extension anzugeben. Betätigt der Anwender hier nur die Eingabetaste ↵(leerer String), dann trägt die Software automatisch den über das Setup (siehe Kommando **`LAYDEFLIBRARY`** in **BSETUP**-Beschreibung) definierten Default-Layoutbibliotheksnamen ein.

Auf die Abfrage nach dem Layout Elementnamen ist der Name der zu generierenden Netzliste bzw. des zu erstellenden Layouts einzutragen (beim ersten **Packager**-Lauf frei wählbar). Betätigt der Anwender hier nur die Eingabetaste ↵w (leerer String), dann trägt die Software automatisch den über das Setup (siehe Kommando **`LAYDEFELEMENT`** in **BSETUP**-Beschreibung) definierten Default-Layoutelementnamen ein.

Nach der Eingabe des Layout Elementnamen arbeitet der **Packager** nacheinander alle in der Projektdatei enthaltenen Stromlaufplanblätter ab und generiert in der Projektdatei unter dem Layout Elementnamen eine Netzliste. Unter dem Projektdateinamen wird mit der Extension `.fre` eine Freelist ausgegeben. Die Freelist wird vom System nicht weiter benötigt und ist zur Auswertung durch den Benutzer bestimmt (Anzeige nicht angeschlossener Pins, Statistik).

Nach erfolgreichem **Packager**-Lauf erscheint die Meldung **`Es wurden keine Fehler festgestellt.`**. Der Anwender gelangt anschließend durch Betätigung einer beliebigen Taste wieder in das Hauptmenü. Sollte es zu Fehlermeldungen kommen, dann deutet dies in der Regel darauf hin, dass die spezifizierte Layoutbibliothek falsche oder fehlende Zuweisungen von Stromlauf- zu Layoutsymbolen aufweist. In diesem Fall sind vor einem neuerlichen **Packager**-Lauf mit Hilfe einer entsprechenden **LOGLIB**-Datei und dem Utilityprogramm **LOGLIB** die richtigen logischen Bauteilreferenzen in die Layoutbibliothek einzutragen.

Backannotationanforderungen

Falls im Layout für die Backannotation relevante Modifikationen wie Pin-/Gate-Swaps oder Bauteilumbenennungen vorgenommen wurden, wird beim Speichern des Layouts ein spezieller Datenbankeintrag erzeugt, der im **Packager** ausgewertet wird, um den Anwender ggf. über eine Bestätigungsabfrage auf die Notwendigkeit zur Durchführung der Backannotation vor dem nächsten **Packager**-Lauf hinzuweisen. Wird dieser Hinweis ignoriert und trotzdem ein **Packager**-Lauf durchgeführt, dann werden die Netzlistenänderungen aus dem Layout verworfen.

Attributtransfer

Die Namen der SCM-Symbolpins werden vom **Packager** automatisch auf das Pinattribut **`$llname`** übertragen und können im Layout durch die Definition entsprechender Texte auf Padstackebene visualisiert werden.

Das Pinattribut **`$nettype`** wird vom **Packager** automatisch von Pins auf die angeschlossenen Netze übertragen. Sind an einem Netz Pins mit unterschiedlichen Attributwerteinträgen für **`$nettype`** angeschlossen so wird für das Netz der Wert **`mixed`** eingetragen.

Das Pinattribut **`$drcblk`** wird vom **Packager** automatisch von Pins auf die angeschlossenen Netze übertragen. Der Attributwerteintrag für **`$drcblk`** addressiert in **BAE HighEnd** einen Parameterblock mit Entwurfsregeln, der damit an das entsprechende Netz zugewiesen wird.

ERC (Electrical Rule Check)

Der **Packager** erlaubt nun über das Pinattribut **`$pintype`** eine Plausibilitätsprüfung für die im Schaltplan vorgenommenen Verbindungen zwischen Pins verschiedener Typen. Die Pinattribute werden zweckmässigerweise in den logischen Definitionen der Symbole fest für die einzelnen Layoutbauteilpins vergeben. Unterstützt werden die folgenden Pintypen:

`$pintype`	**Pintyp**
`in`	Eingabe-Pin
`out`	Ausgabe-Pin
`bidi`	Bidirektionaler Anschluss
`anl`	Analoger Anschluss
`sup`	Stromversorgungsanschluss

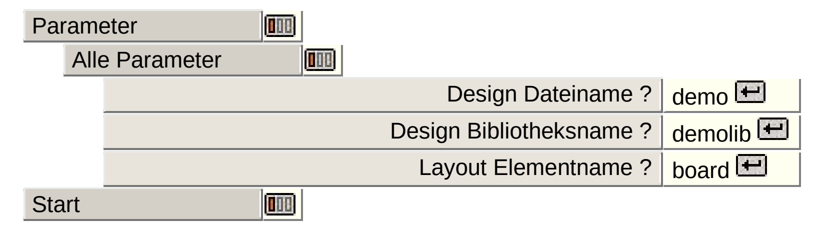
Der ERC überprüft für Netze mit mindestens einem Eingang, ob an diesem Netz ein normaler Ausgang, ein bidirektionaler Anschluss oder ein Versorgungsspannungspin vorhanden ist und gibt ggf. die Warnmeldung **`Netz 'Netzname' hat nur Eingaenge!`** aus. Außerdem überprüft der ERC, ob an einem normalen Ausgang ein anderer Ausgang, ein bidirektionaler Anschluss oder ein Versorgungsspannungspin angeschlossen ist und gibt ggf. die Warnmeldung **`Treiber-Kollision auf Netz 'Netzname'!`** aus.

3.2.4 Beispiel

Im Folgenden soll ein **Packager**-Lauf für den in Kapitel 2 in der Projektdatei **demo.ddb** erstellten Stromlaufplan durchgeführt werden. Wechseln Sie hierzu in das Verzeichnis, in dem **demo.ddb** abgelegt ist.

Fehlerhafter Packager-Lauf

Starten Sie BAE, rufen Sie wie in Kapitel 3.2.1 beschrieben den **Packager** auf, und versuchen Sie mit den folgenden Kommandos die in **demo.ddb** enthaltene logische Netzliste unter Verwendung der Bibliothek **demolib.ddb** in eine physikalische Netzliste mit dem Namen **board** zu transferieren:



Der **Packager**-Lauf wird nicht erfolgreich sein, und auf dem Bildschirm erscheinen folgende Meldungen:

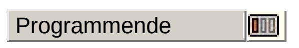
```
================
BARTELS PACKAGER
================

Design Dateiname ........: 'demo'
Bibliothek Dateiname ....: 'demolib'
Layout Elementname ......: 'board'
Aktiver Planname ........: 'sheet1'.
FEHLER : Bauteil 'cd4081' nicht in Bibliothek!
Abbruch ohne Veraenderung der Datenbank.
```

Die Meldung **Abbruch ohne Veränderung der Datenbank** erscheint immer dann am Ende des **Packager**-Protokolls, wenn der **Packager**-Lauf nicht ordnungsgemäß beendet werden konnte.

Unter Windows besitzt das Textfenster zur Ausgabe der durch den **Packager** erzeugten Warnungen und Fehlermeldungen Scrollbars. Dies ermöglicht bei längeren Protokollen das Blättern in der gesamten Liste der Meldungen und erspart somit den Umweg über die Ansicht der Datei **bae.log** (siehe unten).

Betätigen Sie jetzt die Eingabetaste, um wieder in das **Packager**-Hauptmenü zurückzugelangen, und beenden Sie den **AutoEngineer**:



Sie befinden sich nun wieder auf Betriebssystemebene. Das **Packager**-Protokoll, das zuvor auf dem Bildschirm sichtbar war, wurde auch in der Datei **bae.log** (im aktuellen Verzeichnis) abgelegt. Sie können sich diese Datei mit Ihrem Editor ansehen, oder auf Drucker ausgeben, um die darin enthaltene Fehlerliste ggf. zu interpretieren. Der **Packager** beendete die Bearbeitung mit der folgenden Fehlermeldung:

```
FEHLER : Bauteil 'cd4081' nicht in Bibliothek!
```

Diese Fehlermeldung besagt, dass für das Stromlaufsymbol **cd4081** kein Eintrag in der logischen Bibliothek (**demolib.ddb**) enthalten ist. **cd4081** ist das in Kapitel 2.2.2 neu erstellte Stromlaufsymbol, das auf dem Stromlaufblatt **sheet1** der Projektdatei **demo.ddb** mehrfach verwendet wurde. Das Symbol **cd4081** stellt lediglich eines der vier Einzelgatter des kompletten Bausteins CD4081 dar. Die logischen Pins des Stromlaufsymbols wurden mit **A**, **B** und **Y** benannt. Dies ist auch alles, was dem System zu diesem Bauteil bekannt ist. Dem **Packager** fehlt also die Information, welchem Layoutsymbol das Stromlaufsymbol **cd4081** zuzuordnen ist. Außerdem fehlt dem System auch die Zuordnung der logischen Pins zu den physikalischen Anschlüssen, die Information über feste Anschlüsse zur Stromversorgung, sowie die Definition der Pin- und Gattervertauschbarkeit.

Packagerfehlerlokalisierung im Schaltplan

Tritt in den menügesteuerten Versionen des **Packagers** beim **Packager**-Lauf ein Fehler auf, der eindeutig einem Schaltplanblatt zuzuordnen ist, so wird dieses Schaltplanblatt beim anschließenden Wechsel in den **Schaltplaneditor** automatisch zur Bearbeitung in den Speicher geladen. Zusätzlich wird nach Möglichkeit ein Zoom Fenster auf das erste an einem Fehler beteiligte Schaltplansymbol durchgeführt.

Korrektur bzw. Vervollständigung der logischen Bibliothek

Um einen fehlerfreien **Packager**-Lauf für den Job **`demo.ddb`** zu ermöglichen, ist die fehlende Information über die Zuordnung des Stromlaufsymbols **`cd4081`** zum entsprechenden Layoutsymbol in eine **LOGLIB**-Datei einzutragen. Diese Datei ist dann mit Hilfe des Programms **LOGLIB** in die für den **Packager**-Lauf zu verwendende Bibliotheksdatei **`demolib.ddb`** einzuspielen.

Erstellen Sie also zunächst mit Ihrem Editor die ASCII-Datei **`cd4081.def`** mit der in Abbildung 3-2 angegebenen Loglib-Definition als Inhalt.

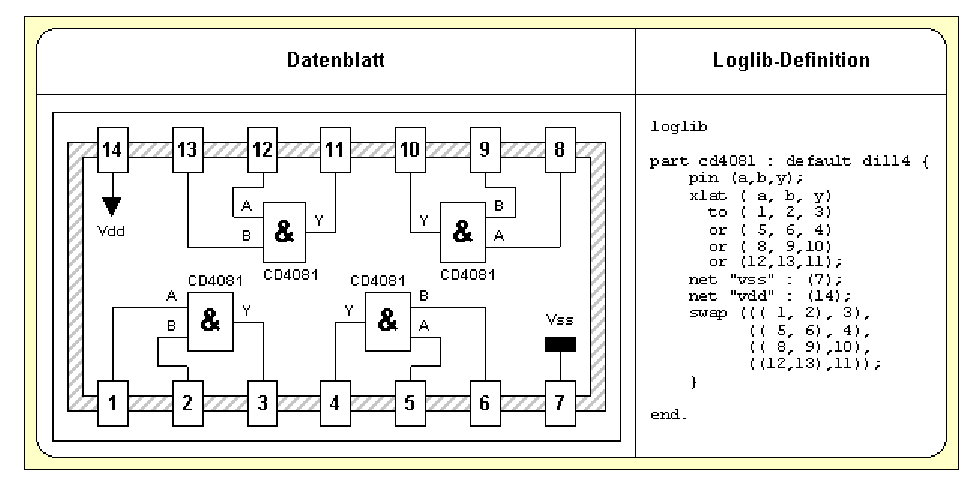


Abbildung 3-2: Datenblatt für Bauteil CD4081 mit Loglib-Definition

Die Syntax für die durch das Programm **LOGLIB** umzusetzenden Dateien ist im Kapitel 7.11> ausführlich beschrieben. An dieser Stelle sei nur der Inhalt der **LOGLIB**-Definition aus Abbildung 3-2 näher erläutert.

Die Schlüsselwörter **`loglib`** und **`end.`** kennzeichnen den Dateianfang bzw. das Dateiende.

Das Kommando **`part`** definiert die Zuordnung des Stromlaufsymbols zum Layoutsymbol. In obigem Beispiel gibt das Schlüsselwort **`default`** an, dass diese Zuweisung nicht zwingend ist. Sie kann durch eine entsprechende Wertzuweisung an das Attribut **`$plname`** (sofern im Stromlaufsymbol definiert) im Schaltplan geändert werden, um z.B. ein SMD-Gehäuse anstelle des DIL-Gehäuses einzusetzen.

Das Kommando **`pin`** definiert die logischen Pins **`A`**, **`B`** und **`Y`** des Stromlaufsymbols. Achten Sie darauf, dass diese Definition mit den bei der Symbolerstellung vergebenen Pinbezeichnungen übereinstimmt (andernfalls erzeugt der **Packager** entsprechende Fehlermeldungen).

Die **`xlat`**-Anweisung ist das Transformationskommando zur Definition der Gatter bzw. zur Zuweisung der logischen zu den physikalischen Pins.

Mit dem **`swap`**-Kommando wird die Pin- und Gattervertauschbarkeit definiert. In obigem Beispiel sind die vier Einzelgatter untereinander vertauschbar, und es besteht Pinvertauschbarkeit für die Eingänge der Gatter.

Die beiden **`net`**-Kommandos schließlich geben die am Bauteil fest definierten Anschlüsse zur Stromversorgung (Netz **`Vss`** am physikalischen Pin **`7`**, Netz **`Vdd`** an Pin **`14`**) an.

Übernahme der logischen Bibliotheksdefinition

Spielen Sie nun mit Hilfe des Programms **LOGLIB** die Informationen aus der soeben erstellten **LOGLIB**-Datei **`cd4081.def`** in die Bibliotheksdatei **`demolib.ddb`** ein:

```
> loglib cd4081 demolib ↵
```

Wenn Sie bei der Erstellung der **LOGLIB**-Datei keine Fehler gemacht haben, dann sollte **LOGLIB** folgende Meldungen ausgeben:

```
===================================
BARTELS LOGICAL LIBRARY MAINTENANCE
===================================

Es wurden keine Fehler festgestellt.
```

Nun enthält die Datei **`demolib.ddb`** alle für einen erfolgreichen **Packager**-Lauf zur Umsetzung der Projektdatei **`demo.ddb`** notwendigen Informationen (vergleichen Sie hierzu auch den Inhalt der bereits in **`demolib.ddb`** eingespielten **LOGLIB**-Datei **`demolib.def`**).

Abfrage/Anzeige logischer Bibliotheksdefinitionen

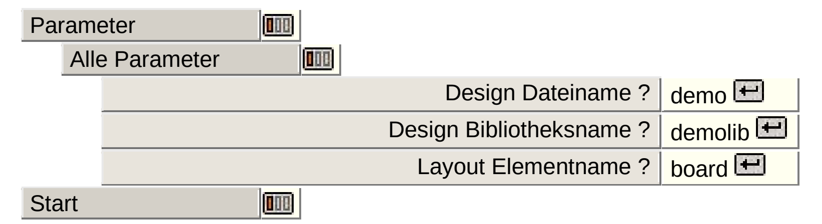
Das Menü Symbole des **Schematic Editors** enthält die Funktion Symbollogik zeigen zur Anzeige der logischen Bauteildefinitionen selektierbarer Symbole des aktuell geladenen Stromlaufplans.

Es erfolgt eine Dekodierung der mit dem Utilityprogramm **LOGLIB** erstellten internen logischen Bibliotheksdefinition sowie die Anzeige der derselben im ASCII-Format. Die Kopfzeile dieser Anzeige enthält die Angabe darüber, ob die Logische Bibliotheksdefinition in der aktuellen Projektdatei (Projekt; hierin wird zuerst gesucht) oder in der Default-Layoutbibliothek (Bibliothek) gefunden wurde. Der Name der Default-Layoutbibliothek ergibt sich aus dem über das **`LAYDEFLIBRARY`**-Kommando des Utilityprogramms **BSETUP** eingetragenen Pfadnamen (siehe hierzu auch Kapitel 7.2).

Die Funktion Symbollogik anzeigen gibt in der Statuszeile die Fehlermeldung **`Datei nicht gefunden!`** aus, wenn der Zugriff auf die Default-Layoutbibliothek fehlschlägt. Die Fehlermeldung **`Symbollogik-Daten ('<symbolname>') nicht gefunden!`** erscheint, wenn die angeforderte Loglib-Definition weder in der Projektdatei noch in der Default-Bibliothek verfügbar ist.

Fehlerfreier Packager-Lauf

Starten Sie nun erneut den **Packager**, und transferieren Sie mit den folgenden Kommandos die in **demo.ddb** enthaltene logische Netzliste unter Verwendung der Bibliothek **demolib.ddb** in eine physikalische Netzliste mit dem Namen **board**:



Der **Packager** gibt folgende Meldungen auf dem Bildschirm aus:

```
================
BARTELS PACKAGER
================

Design Dateiname ........: 'demo'
Bibliothek Dateiname ....: 'demolib'
Layout Elementname ......: 'board'
Aktiver Planname ........: 'sheet1'.
Aktiver Planname ........: 'sheet2'.
Es wurden keine Fehler festgestellt.
```

Der **Packager**-Lauf wurde erfolgreich beendet (Meldung **Es wurden keine Fehler festgestellt.**), und in der Datei **demo.ddb** wurde eine gepackte, physikalische Netzliste mit dem Namen **board** generiert.

Zusätzlich hat der **Packager** mit dem Namenszusatz **_log** (hier also mit dem Namen **board_log**) auch eine logische Netzliste in der Projektdatei erzeugt. Diese logische Netzliste wird vom System nicht weiter benötigt, kann jedoch z.B. mit Hilfe des Utilityprogramms **USERLIST** (siehe unten) für spezielle Schnittstellen zu Fremdsystem (z.B. in Richtung Simulation) benutzt werden (ein Beispiel hierzu finden Sie in Kapitel 3.4.3).

Auch die vom **Packager** erzeugte Freelist **demo.fre** wird vom System nicht weiter benötigt. Sie sollten mit Hilfe wIhres Editors einen Blick auf die in dieser Datei enthaltenen statistischen Informationen werfen, um z.B. festzustellen, ob in Ihrem Schaltplan Signale mit nur einem Anschluss definiert sind.

Der **Packager** bildet die gepackte, physikalische Netzliste auf den Stromlaufplan ab. Überprüfen Sie dies, indem Sie sich nach dem erfolgreichen **Packager**-Lauf im Stromlauf-Editor die entsprechenden Stromlaufblätter ansehen. Dabei sollten Sie im Beispiel **demo.ddb** insbesondere die vier Gatter **cd4081** auf dem **sheet1** beachten. Alle diese Gatter haben nun denselben Bauteilnamen (da Sie alle in dasselbe Gehäuse gepackt wurden), und anstelle der logischen Pinbezeichnungen (jeweils **A**, **B**, **Y**) sind jetzt die physikalischen Anschlussbezeichnungen (**1**, **2**, **3**, usw.) eingetragen.

Die nächsten Schritte

Um nach einem erfolgreichen **Packager**-Lauf zu einem Layout zu gelangen, ist das Layoutmodul (d.h., der **Layouteditor**) des **Bartels AutoEngineer** aufzurufen. Hier ist in der Projektdatei **demo.ddb** zu der Netzliste **board** ein entsprechendes Layout (also mit dem Elementnamen **board**) zu erzeugen. Nachdem dann über das Menü Parameter der Bibliothekspfad auf die beim **Packager**-Lauf verwendete Bibliothek **demolib.ddb** eingestellt wurde, können die in der Netzliste enthaltenen Bauteile (z.B. durch wiederholtes Anwenden der Funktion Naechstes Bauteil aus dem Menü Bauteile) auf dem Layout platziert werden (siehe Kapitel 4.3.2).

3.2.5 Meldungen

Dieser Abschnitt enthält eine alphabetisch sortierte Auflistung der Fehler-/Warnungs- und Statusmeldungen des **Packagers**. Bei Fehler- und Warnungsmeldungen wurde das führende `FEHLER :` bzw. `WARNUNG :` weggelassen, da einige Meldungen abhängig vom Parameter Fehlerbehandlung als Fehler- oder als Warnungsmeldung ausgegeben werden. Die am Bildschirm erscheinenden Meldungen werden parallel in einer Textdatei `bae.log` mitprotokolliert und können somit auch noch nach Verlassen des **Packagers** eingesehen werden.

Wie bei den einzelnen Fehlermeldungen dokumentiert, wird nach einem Fehler, der einem Schaltplansymbol bzw. seinen logischen Definitionen zugeordnet werden kann, bei unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem betreffendem Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeührt. Bei mehreren Fehlermeldungen während eines **Packager**-Laufes trifft dies nur für die erste einem Symbol zuzuordnende Fehlermeldung zu. Ist das Problem der logischen Definition des Symbols zuzuordnen, so kann diese im Schaltplan mit Symbole / Symbollogik zeigen bzw. Symbole / Symbollogik editieren betrachtet bzw. korrigiert werden. Diese Funktionen berücksichtigen auch, dass der Definitionsname über `$rlname`- und/oder `$rlext`-Attribute auf einen alternativen vom Symbolmakronamen abweichenden Definitionsnamen gelegt sein könnte.

Bei der Korrektur von logischen Definitionen ist zu beachten, dass der **Packager** die logischen Definitionen bei der Bearbeitung aus der Bibliothek in die Projektdatei kopiert und bei folgenden **Packager**-Läufen per Default den in der Projektdatei vorhandenen logischen Definitionen Priorität vor den Definitionen in der Bibliothek gibt. Unter Parameter / Updatemodus) kann durch die Einstellung Definitionsupdate nach Korrekturen in der Bibliothek ein erneutes Kopieren der logischen Definitionen aus der Bibliothek in die Projektdatei erzwungen werden.

Bei Korrekturen in Symbolen ist zu beachten, dass nach dem Platzieren, Löschen oder Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen bzw. geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden modifizierte Pindefinitionen aber erst beim Speichern der Schaltplanblätter übertragen.

[110] ASCII-Datei Schreibfehler! Schreibfehler, die Freipindatei ist fehlerhaft!

Beim Schreiben der zum Projekt gehörenden `.fre`-Datei mit der Auflistung der nicht benutzten Pins, Netzpinanzahlen und Netzzusammenfassungen in mit beliebigen Editoren lesbarer Textform ist ein Fehler aufgetreten. Zu diesem Zeitpunkt ist die Layoutnetzliste bereits erfolgreich geschrieben worden, mit dem Projekt kann also trotz des Fehlers normal weitergearbeitet werden. Die oben genannten Informationen stehen allerdings nicht oder nur unvollständig zu Kontrollzwecken zur Verfügung.

Wahrscheinlichste Fehlerursache ist ein Schreibschutz auf einer aus vorhergehenden **Packager**-Läufen stammenden `.fre`-Datei für das Projekt (wie er z.B. bei einfach von CD auf Festplatte kopierten Sicherungskopien gesetzt wird). Bei Reproduzierbarkeit des Problems ohne Schreibschutz und mit ausreichend freiem Festplattenplatz empfiehlt es sich, die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[012] Abbruch ohne Veraenderung der Datenbank.

Bei der Bearbeitung des Projektes traten ein oder mehrere Fehler auf, die das Erstellen einer korrekten Layoutnetzliste verhinderten. Der Vorgang wurde ohne Schreiben einer Netzliste abgebrochen. Eine ggf. bereits vorhandene alte Layoutnetzliste wurde nicht verändert und kann weiterhin verwendet werden. Die genauen Fehler sollten aus vorhergehenden Fehlermeldungen hervorgehen.

[019] Aktiver Block/Planname ....: '[nnn]blockname' / 'blockplanname'.

Es wird gerade ein Schaltplanblatt einer Blockreferenz eines hierarchischen Schaltplanes bearbeitet. `nnn` ist die laufende Nummer der Blockreferenzen, `blockname` der auf den Unterblöcken und in der logischen Definition zum Blocksymbol angegebene Blockname und `blockplanname` der Name des gerade bearbeiteten Schaltplanblattes (ein hierarchischer Block kann aus mehreren Schaltplanblättern mit gleichem Blocknamen zusammengesetzt sein). Unmittelbar nachfolgende Fehler- und Warnungsmeldungen beziehen sich auf Symbole dieses Blattes.

[018] Aktiver Planname ....: 'planname'.

Es wird gerade das Schaltplanblatt mit dem Namen `planname` bearbeitet. Unmittelbar nachfolgende Fehler- und Warnungsmeldungen beziehen sich auf Symbole dieses Blattes.

[109] Allgemeiner Datenbankfehler!

Beim Zugriff auf eine `DDB`-Datei trat ein nicht näher spezifizierter Fehler auf. Es ist zunächst zu prüfen, ob die Datei nicht evtl. schreibgeschützt ist (wie es z.B. bei einfach von CD auf Festplatte kopierten Sicherungskopien der Fall ist). Bei Reproduzierbarkeit des Problems ohne Schreibschutz und mit ausreichend freiem Festplattenplatz empfiehlt es sich, die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[005] Alternativbibliothek Dateiname : 'alternativbibliotheksname'

Hier wird der Name einer ggf. in den Einstellungen gesetzten Alternativbibliothek dokumentiert. Auf die Alternativbibliothek wird zugegriffen, wenn ein Layoutbauteilmakro oder eine logische Definition weder in der Projektdatei noch in der Standardlayoutbibliothek gefunden werden kann. Layoutbauteilmakros und logische Definitionen werden vom **Packager** in die Projektdatei kopiert, d.h. es ist nach einem erfolgreichen **Packager**-Lauf nicht notwendig, im **Layouteditor** die Bibliothek auf die Alternativbibliothek zu setzen um die in der Netzliste verwendeten Bauteile erfolgreich zu platzieren. Der Parameter Alternativbibliothek wird in der Projektdatei gespeichert und muss bei folgenden **Packager**-Läufen nicht erneut definiert werden.

[076] Attributdaten zu Symbol 'symbolname' Blatt 'planname' nicht gefunden!

Der zu dem auf dem Schaltplanblatt `planname` platzierten Symbol mit dem Namen `symbolname` gehörige Datenbankeintrag mit den Bauteilattributen konnte in der Projektdatei nicht gefunden werden. Vemutlich wurde dieser Datenbankeintrag im **Schaltplaneditor** mit Hilfe der Funktionen Datei / Element loeschen / Bauteil oder Datei / Library Utilities / Elemente loeschen / Bauteil von Hand gelöscht. Diese Funktionen sollten nur mit ässerster Sorgfalt angewendet werden, da die Attributwerte von platzierten Symbolen gelöscht werden können. Zur Bereinigung der Bauteilnamensliste sollte statt dessen die Funktion Symbole / Weitere Funktionen / Namensliste Cleanup verwendet werden, die nur die Bauteilattributeinträge von im Projekt nicht mehr platzierten Symbolen löscht. Nach einem erfolgreichen **Packager**-Lauf werden nicht benützte Bauteildatenbankeinträge automatisch entfernt, ein manueller Cleanup ist in der Regel somit nur notwendig, wenn in einem aus Altprojekten abgeleiteten Projekt wegen fehlenden Definitionen noch kein **Packager**-Lauf möglich ist.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition eines der Teilsymbole durchgeführt.

Zur Behebung des Problems ist das Schaltplanblatt mit dem betroffenen Symbol (siehe "Aktiver Planname") im **Schaltplaneditor** zu laden und erneut zu Speichern. Dabei werden automatisch Bauteilattributdatenbankeinträge für alle auf dem Schaltplanblatt vorhandenen Symbole erzeugt. Für die betroffenen Symbole sind diese zunächst leer und müssen ggf. mit Symbole / Wert(e) zuweisen erneut mit Werten belegt werden.

[014] Aufruf des Programmmoduls fehlgeschlagen!

Der Wechsel in eines der anderen Programmmodule des BAE konnte nicht durchgeführt werden. Dies deutet auf ein Installationsproblem hin. Es wurden entweder `.exe`-Dateien aus dem BAE-Programmverzeichnis entfernt oder es liegt eine Mischinstallation zwischen `.exe`-Dateien der **HighEnd**-Version und `.exe`-Dateien anderer BAE-Versionen vor. So eine Mischinstallation kann z.B. entstehen, wenn von der Demoversion durch Entpacken der Datei `baew32he.zip` zur **HighEnd**-Version gewechselt werden soll und während dem Entpacken BAE-Module aktiv sind. Das Betriebssystem lässt das Überschreiben der aktiven `.exe`-Dateien nicht zu und es bleiben Teile der alten Version installiert.

[049] Bauteil 'layoutbauteilname' ('symbolname') ist ueberbelegt

(Liste:) Log. Bauteil : 'symbolnamen'

Es wurde über das **$rpname**-Attribut versucht mehr Gatter eines Symbolmakrotypes in ein Layoutgehäuse zu packen, als über das **xlat**-Kommando in der logischen Definition aufgeführt wurden. **symbolname** gibt den Namen des Schaltplansymboles an, das die Überbelegung verursacht. In der Liste der **symbolnamen** werden die Namen der Schaltplansymbole aufgegührt, die bereits in das Layoutbauteil gepackt wurden.

Die Überbelegung kann auch dadurch entstehen, dass über das **$rpname**-Attribut zwar die richtige Anzahl Gatter in ein Layoutgehäuse gepackt wird, es aber noch ein gleichnamiges Schaltplansymbol mit dem angeforderten Namen gibt, das keine eigene Namenszuweisung über **$rpname** besitzt. Der **Packager** ist nicht in der Lage eigenständig einen alternativen Layoutbauteilnamen für dieses Symbol zu bilden.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition eines der Teilsymbole durchgeführt.

Zur Behebung des Problems sind die Namenszuweisungen über das **$rpname**-Attribut den tatsächlich vorhandenen Bauteilgattern anzupassen. Bei der Überprüfung der logischen Definitionen ist zu beachten, dass an den betroffenen Symbolen über **$rlname**-bzw. **$rlext**-Attribute alternative vom Symbolmakronamen abweichende Definitionen selektiert sein könnten.

Die gleiche Problematik kann bei Anwendung des **$spname**-Attributes entstehen, das im Gegensatz zum **$rpname**-Attribut aber nur bis zur ersten erfolgreichen Zuweisung eines Schaltplansymboles zu einem Layoutbauteil berücksichtigt wird und im Allgemeinen nur beim Import von Netzlisten aus Fremdsystemen Anwendung findet.

[050] Bauteil 'layoutbauteilname' ('symbolname') selektierte Definition bzw. Gehaeusenamen inkompatibel zu:

Es wurde versucht das Symbol mit dem Namen **symbolname** über ein **$rpname**-, **$spname**-oder **$vgrp**-Attribut in das Layoutbauteil mit dem Namen **layoutbauteilname** zu packen in das bereits die in der nachfolgenden Liste aufgeführten Symbole gepackt wurden. Die logische Definition des Symbols **symbolname** bzw. über ein **$plname**-Attribut vorgenommene Gehäusezuordnung sind nicht kompatibel zu den bereits in das Layoutbauteil gepackten Symbolen.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition eines der Teilsymbole durchgeführt.

Zur Behebung des Problems sind die logischen Definitionen der einzelnen in das Layoutbauteil gepackten Symbole für gleiche Symbolmakros gleich zu wählen, bzw. es ist bei den Symbolen eine einheitliche Gehäusezuweisung vorzunehmen.

[128] Bauteil 'layoutbauteilname' Alternativbauformen durch $plname deaktiviert!

Einem Bauteil, das in der logischen Definition eine Alternativbauformliste besitzt, wurde über das Attribut **$plname** ein fester Gehäusename zugewiesen. Die Alternativbauformliste wurde daher für dieses Bauteil deaktiviert.

Es ist zu überprüfen, ob das verwendete Symbol und die logische Definition zueinander passen, da eine gemischte Verwendung von **$plname**-Attributen und Alternativbauformlisten in der logischen Definition nicht vorgesehen ist. Es ist möglich, die Alternativbauformliste aus der logischen Definition zu entfernen und statt dessen auf Symbolebene in der Form **[gehaeuse1,gehaeuse2,...]** dem **$plname**-Attribut als Defaultwert zuzuweisen. Dann kann die als Attributwert beim ersten Platzieren des Symbols gesetzte Alternativbauformliste mit Wert(e) zuweisen wahlweise in eine Einzelbauform umeditiert werden, ohne dass im **Packager** Warnungsmeldungen erscheinen.

[089] Bauteil 'layoutbauteilname' enthaelt unbenutzte Gatter (Pin 'layoutpinname')!

Im Layoutbauteil mit dem Namen **layoutbauteilname** sind nicht für alle in **xlat**-Kommando(s) der logischen Definition(en) aufgeführten Gatter Entsprechungen im Schaltplan vorhanden, d.h. es sind noch Gatter zur Benutzung frei.

Zur genaueren Spezifikation der freien Gatter wird jeweils der Namen des ersten im **xlat**-Kommando aufgeführten Pins des Layoutbauteils aufgeführt.

Dies ist nur eine Warnungsmeldung, die nicht zum Abbruch des **Packager**-Laufes führt. Es ist jedoch zu beachten, dass bei bestimmten Typen von Bausteinen der undefinierte Zustand der nicht angeschlossenen Eingänge von unbenutzten Gattern zu einem undefinierten Verhalten des gesamten Bausteines führen kann. In so einem Fall empfiehlt es sich im Schaltplan Symbole für die nicht benutzten Gatter zu platzieren und die Eingänge auf ein definiertes Potential zu legen. Da vom **Packager** auch nicht benutzte Gatter mit der Bauteilkennung versehen werden, kann es notwendig sein für neu platzierte Gatter das Packen in das bereits bestehende Bauteil durch Setzen des **$rpname**-Attributs auf den **layoutbauteilnamen** zu erzwingen.

[033] Bauteil 'layoutbauteilname' mehrfach definiert!

Diese Fehlermeldung kann als Folgefehler der Meldung **Bauteil 'layoutbauteilname' ('symbolname') ist ueberbelegt (Liste:) Log. Bauteil : 'symbolnamen'** auftreten. Sollte die Meldung einzelen auftreten, deutet dies auf ein noch nicht bekanntes Problem im **Packager** hin. Lässt sich das Problem nicht mit den Hinweisen zur Meldung **Bauteil 'layoutbauteilname' ('symbolname') ist ueberbelegt (Liste:) Log. Bauteil : 'symbolnamen'** lösen, empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[080] Bauteil 'layoutbauteilname' verschiedene Gehaeusezuweisungen ('gehaeusename1'<>'gehaeusename2')!

Dem Layoutbauteil mit dem Namen **layoutbauteilname** wurden über **$plname**-Attribute seiner Teilsymbole die unterschiedlichen Gehäuse mit den Namen **gehaeusename1** und **gehaeusename2** zugewiesen.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster

[082] Bauteil 'layoutbauteilname' verschiedene Werte fuer 'attributname' ('wert1'<>'wert2')!

Dem Layoutbauteil mit dem Namen **layoutbauteilname** wurden für das Attribut **attributname** die beiden unterschiedlichen Attributwerte **wert1** und **wert2** zugewiesen. Dies kann dadurch zustande kommen, dass das Layoutbauteil aus mehreren Symbolen zusammengesetzt ist, die zusammengepackt werden und die unterschiedliche Attributwerte besitzen. Als Quelle für die unterschiedlichen Attributwerte des Layoutbauteils kommen das oder die zugehörigen Schaltplansymbole, ein **newattr**-Kommando in der oder den logischen Definitionen der Symbole oder an das oder die Symbole angehängte Tagsymbole mit Attributzuweisungen in Betracht. Dies ist nur eine Warnungsmeldung, der **Packager** setzt die Bearbeitung des Projektes fort und verwendet **wert1** als Attributwert, der **wert2** wird verworfen.

Zur Behebung des Problems sind die Attributzuweisungen der Symbole bzw. der logischen Definitionen anzugleichen.

[084] Bauteil 'layoutbauteilname' verschiedene Werte fuer 'attributname'/variantennummer('wert1'<>'wert2')!

Dem Layoutbauteil mit dem Namen **`layoutbauteilname`** wurden in der Variante mit der Nummer **`variantennummer`** für das Attribut **`attributname`** die beiden unterschiedlichen Attributwerte **`wert1`** und **`wert2`** zugewiesen. Dies kann dadurch zustande kommen, dass das Layoutbauteil aus mehreren Symbolen zusammengesetzt ist, die zusammengepackt werden und die unterschiedliche Attributwerte besitzen. Als Quelle für die unterschiedlichen Attributwerte des Layoutbauteils kommen das oder die zugehörigen Schaltplansymbole, ein **`newattr`**-Kommando in der oder den logischen Definitionen der Symbole oder an das oder die Symbole angehängte Tagsymbole mit Attributzuweisungen in Betracht. Dies ist nur eine Warnungsmeldung, der **Packager** setzt die Bearbeitung des Projektes fort und verwendet **`wert1`** als Attributwert in der betreffenden Variante, der **`wert2`** wird für diese Variante verworfen.

Zur Behebung des Problems sind die Attributzuweisungen der Symbole bzw. der logischen Definitionen für die betreffende Variante anzugleichen.

[090] Bauteil 'layoutbauteilnamen' mainpart nicht benutzt!

Diese Warnungsmeldung wird ausgegeben, wenn bei einem in den logischen Definitionen aus **`mainpart`**/**`subpart`** zusammengesetzten Baustein für das Layoutbauteil mit dem Namen **`layoutbauteilnamen`** das zu der **`mainpart`**-Definition gehörige Bauteil im Schaltplan nicht verwendet wird. Abhängig von den logischen Definitionen birgt dies das Risiko, dass wichtige für die Funktion des Bausteins benötigte Pins nicht angeschlossen sind. Das betreffende Symbol für das **`mainpart`** sollte auf dem Schaltplan platziert und dessen Eingangspins ggf. auf ein definiertes Potential gelegt werden. Da vom **Packager** auch nicht benutzte Gatter mit der Bauteilkennung versehen werden, kann es notwendig sein für neu platzierte Symbol das Packen in das bereits bestehende Bauteil durch Setzen des **`$rpname`**-Attributs auf den **`layoutbauteilnamen`** zu erzwingen.

[004] Bibliothek Dateiname ....: 'bibliotheksname'

Hier wird der Name der beim **Packager**-Lauf verwendeten Standardbibliothek dokumentiert. Auf die Standardbibliothek wird zugegriffen, wenn ein Layoutbauteilmakro oder eine logische Definition nicht in der Projektdatei gefunden werden kann. Layoutbauteilmakros und logische Definitionen werden vom **Packager** in die Projektdatei kopiert, d.h. es ist nach einem erfolgreichen **Packager**-Lauf nicht notwendig im **Layouteditor** die gleiche Layoutbibliothek zu setzen um die in der Netzliste verwendeten Bauteile erfolgreich zu platzieren. Der Parameter Bibliotheksname wird in der Projektdatei gespeichert und muss bei folgenden **Packager**-Läufen nicht erneut definiert werden.

[044] Bibliotheksteilname ($plname) fuer 'symbolname' ist nicht aenderbar. Kein default-Kommando in logischer Definition!

An dem Symbol mit dem Namen **`symbolname`** ist das Attribut **`$plname`** zur Selektion einer alternativen Gehäuseform gesetzt, in der logischen Definition für das Symbol ist aber kein **`default`**-Kommando angegeben. Das **`default`**-Kommando signalisiert, dass alternative Gehäuseformen für dieses Symbol über das Attribut **`$plname`** gesetzt werden dürfen. Ohne **`default`**-Kommando gilt der in der logischen Definition vorgebenene Gehäusename als feste Vorgabe für das Symbol.

Zur Behebung des Problems ist das **`default`**-Kommando in der logischen Definition des Symbols nachzutragen oder das **`$noplc`**-Attribut zurückzusetzen und ggf. DANACH der **`$plname`**-Text vom Symbol zu entfernen um weitere Fehlzuweisungen zu verhindern.

[027] Blockname von Netz 'netzname' ist zu lang!

Bei der Layoutnetznamensbildung für das lokale Netz **netzname** aus einem hierarchischen Schaltplanblock wurde die maximale Namenslänge von 40 Zeichen überschritten. Zur Bildung des Layoutnetznamens werden die lokalen Netznamen eines mehrfach verwendbaren hierarchischen Schaltplanblattes mit einem Prefix der Form **[lblocknummer]** versehen um eine Unterscheidung zwischen den lokalen Netzen mehrerer Blockreferenzen zur erhalten. Lokale Netze auf Top-Level Blättern erhalten den festen Prefix **[lt]**. Je nach Anzahl von Blöcken im Design und der damit verbundenen Länge der **blocknummer** muss man sich daher bei lokalen Netznamen auf eine Namenslänge von 34 bis 36 Zeichen beschränken. Für Einzel-Sub-Blöcke ist der Namensprefix **[lsblockname]**. Die Beschränkung der Netznamenslänge hängt hier wesentlich von der Länge des **blocknamens** ab.

Zur Problembehebung ist der Netzname im Schaltplan durch Umbenennen des Netzlabels entsprechend zu kürzen.

[026] Blockname von Port 'modulportname' ist zu lang!

Bei Bildung des internen Portnetznamens für den Modulport **modulportname** eines hierarchischen Schaltplanblockes wurde die maximale Namenslänge von 40 Zeichen überschritten. Zur Bildung der internen Portnetznamen werden die Modulportname eines mehrfach verwendbaren hierarchischen Schaltplanblattes mit einem Prefix der Form **[mblocknummer]** versehen um eine Unterscheidung zwischen den internen Portnetzen mehrerer Blockreferenzen zur erhalten. Lokale Modulports auf Top-Level Blättern erhalten den festen Prefix **[mt]**. Je nach Anzahl von Blöcken im Design und der damit verbundenen Länge der **blocknummer** muss man sich daher bei Modulportnamen auf eine Namenslänge von 34 bis 36 Zeichen beschränken. Für Einzel-Sub-Blöcke ist der Namensprefix **[msblockname]**. Die Beschränkung der Modulportnamenslänge hängt hier wesentlich von der Länge des **blocknamens** ab.

Zur Problembehebung ist der Modulportname im Schaltplan durch Umbenennen des Portlabels und Blocksymbolpins entsprechend zu kürzen.

[028] Blockname von Symbol 'symbolname' ist zu lang!

Bei der Layoutbauteilnamensbildung für das Symbol **symbolname** aus einem hierarchischen Schaltplanblock wurde die maximale Namenslänge von 40 Zeichen überschritten. Zur Bildung des Layoutbauteilnamens wird der Symbolname oder das **$rpname**-Attribut von Symbolen auf hierarchischen Schaltplanblättern mit einem Prefix der Form **[pblocknummer]** versehen um eine Unterscheidung zwischen den Layoutbauteilen mehrerer Blockreferenzen zur erhalten. Je nach Anzahl von Blöcken im Design und der damit verbundenen Länge der **blocknummer** muss man sich daher bei Symbolen auf mehrfach verwendbaren hierarchischen Schaltplanblättern auf eine Namenslänge von 34 bis 36 Zeichen beschränken.

Zur Problembehebung ist der betreffende Symbolname im Schaltplan durch Umbenennen des Symbols zu kürzen.

[131] Bzw. kein Pin 'layoutbauteilpinname' auf Bauteilmakro 'layoutbauteilmakroname' vorhanden!

Der in den vorhergehenden Meldungszeilen aufgelistete Symbolpin ist in der logische Definition für das Symbol im **pin**-Kommando nicht aufgelistet bzw. falls kein **pin**-Kommando in der logischen Definition vorhanden ist war keine 1:1-Namenszuordnung zu einem Layoutbauteilpin **layoutbauteilpinname** des verwendeten Layoutbauteilmakros **layoutbauteilmakroname** möglich, da es auf dem Layoutbauteilmakro keinen Pin mit diesem Namen gibt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der fragliche Pin mit einem **xlat**-Kommando der logischen Definition einem existierenden Layoutbauteilpin zuzuordnen oder in das **pin**-Kommando der logischen Definition aufzunehmen oder auf dem Layoutbauteilmakro zu platzieren oder auf dem Schaltplansymbol in einen in den **pin**-Kommandos definierten Namen umzubenennen. Es ist zu beachten, dass nach dem Platzieren/Löschen/Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen/geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden die Daten aber erst beim Speichern der Schaltplanblätter übertragen.

[134] Datei 'dateiname' Lesezugriff nicht erlaubt!

Der Lesezugriff auf die DDB- bzw. DAT-Datei mit dem Namen **dateiname** wurde verweigert. Dies dürfte an mangelnden Zugriffsrechten des Benutzers für diese Datei liegen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzten Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[096] Datei 'dateiname' ist keine Datenbank!

Die über den Namen **dateiname** spezifizierte Datei ist nicht im gültigen BAE-DDB-Dateiformat. Diese Datei wurde vermutlich nicht mit dem BAE erstellt oder von einem anderen Programm überschrieben. Eine automatische Carriage-Return/Linefeed-Konvertierung bei der Dateiübertragung per E-Mail oder beim Netzwerktransfer zwischen unterschiedlichen Betriebssystemen ist ebenfalls als Fehlerursache denkbar. Zur Überprüfung kann versucht werden die betroffene Datei in einen Texteditor zu laden und aus den ersten Zeichen auf den Inhalt bzw. den Dateityp zu schliessen. Eine gültige **DDB**-Datei beginnt mit der Zeichenkette **DTYPE001**.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[136] Datei 'dateiname' ist schreibgeschuetzt!

Der Schreibzugriff auf die DDB- bzw. DAT-Datei mit dem Namen **dateiname** wurde verweigert. Dies dürfte an einem gesetzten Schreibschutz für die Datei oder mangelnden Zugriffsrechten des Benutzers für diese Datei liegen. In Frage kommen insbesondere von CD-ROMs kopierte Dateien, da beim Kopiervorgang solcher Dateien der Schreibschutz beibehalten wird.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzten Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[104] Datei 'dateiname' ist zur Programmversion inkompatibel!

Die über den Namen **dateiname** spezifizierte DDB- bzw. DAT-Datei ist mit einer neueren Version des BAE abgespeichert worden als die für den **Packager**-Lauf verwendete und kann daher Einträge enthalten, die von der verwendeten **Packager**-Version noch nicht unterstützt werden.

Zur Problembehebung ist mindestens auf die gleiche BAE-Versionsnummer upzudaten, die beim Speichern der Datei verwendet wurde. Neuere Versionen können problemlos genutzt werden. Falls die Bearbeitung im Kundenauftrag durchgeführt wird ist jedoch zu bedenken, dass der Kunden für den Rücktransfer der Daten wieder mindestens die gleiche BAE-Versionsnummer benötigt.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[108] Datei 'dateiname' nicht gefunden!

Ein Lesezugriff auf die DDB- bzw. DAT-Datei mit dem Namen **dateiname** konnte nicht durchgeführt werden, da die Datei nicht exisitiert, bzw. keine Zugriffsrechte auf die Datei bzw. das Verzeichnis der Datei vorhanden sind.

Zur Problembehebung sind die Zugriffsrechte entsprechend zu ändern oder die Dateinamen in den **Packager**-Parametern zu ändern.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[133] Datei Lesezugriff nicht erlaubt!

Der Lesezugriff auf eine nicht näher spezifizierte DDB- bzw. DAT-Datei wurde verweigert. Dies dürfte an mangelnden Zugriffsrechten des Benutzers für diese Datei liegen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzten Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[095] Datei ist keine Datenbank!

Eine nicht näher spezifizierte Datei ist nicht im gültigen BAE-DDB-Dateiformat. Diese Datei wurde vermutlich nicht mit dem BAE erstellt oder von einem anderen Programm überschrieben. Eine automatische Carriage-Return/Linefeed-Konvertierung bei der Dateiübertragung per E-Mail oder zwischen unterschiedlichen Betriebssystemen ist ebenfalls als Fehlerursache denkbar. Zur Überprüfung kann versucht werden, die betroffene Datei in einen Texteditor zu laden und aus den ersten Zeichen auf den Inhalt bzw. den Dateityp zu schliessen. Eine gültige **DDB**-Datei beginnt mit der Zeichenkette **DTYPE001**.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[135] Datei ist schreibgeschuetzt!

Der Schreibzugriff auf eine nicht näher spezifizierte DDB- bzw. DAT-Datei wurde verweigert. Dies dürfte an einem gesetzten Schreibschutz für die Datei oder mangelnden Zugriffsrechten des Benutzers für diese Datei liegen. In Frage kommen insbesondere von CD-ROMs kopierte Dateien, da beim Kopiervorgang solcher Dateien der Schreibschutz beibehalten wird.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzten Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[103] Datei ist zur Programmversion inkompatibel!

Eine nicht näher spezifizierte DDB- bzw. DAT-Datei ist mit einer neueren Version des BAE abgespeichert worden als die für den **Packager**-Lauf verwendete und kann daher Einträge enthalten, die von der verwendeten **Packager**-Version noch nicht unterstützt werden.

Zur Problembehebung ist mindestens auf die gleiche BAE-Versionsnummer upzudaten, die beim Speichern der Dateien verwendet wurde. Neuere Versionen können problemlos genutzt werden. Falls die Bearbeitung im Kundenauftrag durchgeführt wird ist jedoch zu bedenken, dass der Kunden für den Rücktransfer der Daten wieder mindestens die gleiche BAE-Versionsnummer benötigt.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[107] Datei nicht gefunden!

Ein Lesezugriff auf eine nicht näher spezifizierte DDB- bzw. DAT-Datei konnte nicht durchgeführt werden, da die Datei nicht exisitiert, bzw. keine Zugriffsrechte auf die Datei bzw. das Verzeichnis der Datei vorhanden sind. Da im **Packager** für alle Dateizugriffe eigene Fehlermeldungen mit Dateinamen definiert sind, deutet diese Meldung auf ein internes Problem im **Packager** hin.

Zur Problembehebung kann versucht werden den **Packager** zu verlassen und erneut aufzurufen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[102] Datenbank Limit ueberschritten!

Beim Zugriff auf eine nicht näher spezifizierte DDB- bzw. DAT-Datei wurde ein internes Limit für Blockgrössen oder verschachtelte Dateisektionen überschritten. Dies sollte im normalen **Packager**-Betrieb nicht vorkommen und könnte ein Folgefehler aus vorhergehenden **Packager**-Läfen der gleichen **Packager**-Sitzung sein oder auf Datenfehler in der bearbeiteten DDB- bzw. DAT-Datei hindeuten.

Zur Problembehebung kann versucht werden den **Packager** zu verlassen und erneut aufzurufen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[094] Datenbank Schreib-/Lesefehler (dateiname)!

Beim Zugriff auf die DDB- bzw. DAT-Datei mit dem Namen **dateiname** ist ein Fehler aufgetreten. Dies kann an einem Fehler auf dem Datenträger, einem vollen Datenträger oder mangelnden Zugriffsrechten des Benutzers für diese Datei liegen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[093] Datenbank Schreib-/Lesefehler!

Beim Zugriff auf eine nicht näher spezifizierte DDB- bzw. DAT-Datei ist ein Fehler aufgetreten. Dies kann an einem Fehler auf dem Datenträger, einem vollen Datenträger oder mangelnden Zugriffsrechten des Benutzers für diese Datei liegen.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[092] Datenbankdatei 'dateiname' kann nicht angelegt werden!

Die DDB- bzw. DAT-Datei mit dem Namen **dateiname** konnte nicht angelegt werden. Dies kann an mangelnden Zugriffsrechten auf das Verzeichnis der Datei liegen. Da im normalen **Packager**-Betrieb DDB- und DAT-Dateien nur geändert und nicht neu angelegt werden, deutet dies auf einen während dem **Packager**-Lauf aufgetreteten Fehler auf dem Datenträger oder das Fremdverschieben/Löschen der DDB- bzw. DAT-Datei während des **Packager**-Laufes hin (z.B. durch einen anderen Anwender im Netzwerk).

In Betracht kommt für diese Fehlermeldung nur die Projektdatei, da auf andere DDB- und DAT-Dateien nur lesend zugegriffen wird.

[091] Datenbankdatei kann nicht angelegt werden!

Eine nicht näher spezifizierte DDB- bzw. DAT-Datei konnte nicht angelegt werden. Dies kann an mangelnden Zugriffsrechten auf das Verzeichnis der Datei liegen. Da im normalen **Packager**-Betrieb DDB- und DAT-Dateien nur geändert und nicht neu angelegt werden, deutet dies auf einen während dem **Packager**-Lauf aufgetreteten Fehler auf dem Datenträger oder das Fremdverschieben/Löschen der DDB- bzw. DAT-Datei während des **Packager**-Laufes hin (z.B. durch einen anderen Anwender im Netzwerk).

In Betracht kommt für diese Fehlermeldung nur die Projektdatei, da auf andere DDB- und DAT-Dateien nur lesend zugegriffen wird.

[106] Datenbankeintrag 'elementname' nicht gefunden!

Das Datenbankelement mit dem Namen **elementname** konnte in einer DDB- bzw. DAT-Datei nicht gefunden werden. Da im **Packager** für alle Elementzugriffe eigene Fehlermeldungen mit Elementnamen definiert sind, deutet diese Meldung auf ein internes Problem im **Packager** hin und die Projektdatei sollte zur näheren Untersuchung an den Bartels-Support gesendet werden.

[105] Datenbankeintrag nicht gefunden!

Ein nicht näher spezifiziertes Datenbankelement konnte in einer DDB- bzw. DAT-Datei nicht gefunden werden. Da im **Packager** für alle Elementzugriffe eigene Fehlermeldungen mit Elementnamen definiert sind, deutet diese Meldung auf ein internes Problem im **Packager** hin und die Projektdatei sollte zur näheren Untersuchung an den Bartels-Support gesendet werden.

[045] Definition 'definitionsname' Hauptbauteil ist undefiniert!

Die von der logischen Definition mit dem Namen `definitionsname` über das `subpart`-Kommando referenzierte logische Definition enthält kein `mainpart`-Kommando. Dies ist nicht erlaubt. Von `subpart`-Kommandos referenzierte logische Definition müssen als `mainpart` deklariert sein.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Behebung des Problems ist die referenzierte logische Definition um das `mainpart` zu ergänzen.

[052] Definition 'definitionsname' Unterbauteile net internal deaktiv!

Das in der vorhergehenden Meldungszeile aufgeführte Symbol mit der logischen Definition `definitionsname` ist das mehr als 32. unterschiedliche `subpart` eines aus `mainpart`/`subpart`-Symbolen zusammengesetzten Bausteins. Für solche Unterbauteile ist das `net internal`-Kommando inaktiv und ggf. angeforderte interne Verbindungen werden nicht in der Netzliste realisiert.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Behebung dieses Problems können die gewünschten Verbindungen in `net internal`-Kommandos des `mainpart` angegeben werden.

[132] Dem Netz 'n.c.' zugewiesene Pins wurden ignoriert.

Diese Meldung am Ende des **Packager**-Laufes signalisiert, dass über `net`-Kommandos dem Netz `n.c.` zugewiesene Pins für die Netzliste ignoriert wurden. Diese Zuordnung ist nützlich, wenn auf der ersten für das Symbol gelisteten Alternativbauform funktionslose Pins vorhanden sind, die auf anderen Bauformen der Alternativbauformliste fehlen. Da der **Packager** normalerweise für jeden Pin automatisch ein Netz anlegt, kommt es ohne Zuordnung zum `n.c.`-Netz nach dem Wechsel zu diesen Alternativbauformen ggf. zu Fehlermeldungen über fehlende Netzlistenpins.

[100] Der Dateiaufbau ist fehlerhaft (dateiname)!

Ein Datenbankelement in der Datei `dateiname` enthält Optionen, die vom **Packager** nicht verstanden werden. Dies kann von einem Fehler auf dem Datenträger oder der Verwendung einer neueren BAE-(Zwischen)Version beim Speichern der Schaltpläne, logischen Definitionen, Layoutbauteilmakros oder Layouts herrühren.

Zur Problembehebung kann versucht werden die logischen Definitionen erneut in das Projekt bzw. die verwendete Bibliothek einzuspielen bzw. die Schaltpläne zu laden und erneut zu speichern. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[099] Der Dateiaufbau ist fehlerhaft!

Ein Datenbankelement in einer nicht näher spezifizierten DDB- bzw. DAT-Datei enthält Optionen, die vom **Packager** nicht verstanden werden. Dies kann von einem Fehler auf dem Datenträger oder der Verwendung einer neueren BAE-(Zwischen)Version beim Speichern der Schaltpläne, logischen Definitionen, Layoutbauteilmakros oder Layouts herrühren.

Zur Problembehebung kann versucht werden die logischen Definitionen erneut in das Projekt bzw. die verwendete Bibliothek einzuspielen bzw. die Schaltpläne zu laden und erneut zu speichern. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[010] Der Dateiname enthaelt ungueltige Zeichen!

Einer der für Projektdatei oder Bibliotheken angegebenen Dateinamen enthält für Dateinamen ungültige Steuerzeichen oder die ebenfalls ungültigen Zeichen **?** und ***** oder der Projektdateiname (Parameter / Design Dateiname) ist nicht gesetzt.

Zur Problembehebung ist der Projektdateiname zu setzen oder es sind Dateinamen zu verwenden, die diese ungültigen Zeichen nicht enthalten.

[116] Der Layoutelementname enthaelt ungueltige Zeichen!

Der unter Parameter / Layoutelementname angegebene Zielname für die Layoutnetzliste enthält für Elementnamen ungültige Steuerzeichen oder die ebenfalls ungültigen Zeichen **?** und ***** oder ist nicht gesetzt.

Zur Problembehebung ist ein anderer Zielname für die Layoutnetzliste zu spezifizieren.

[003] Design Dateiname ....: 'projektdateiname'

Diese Meldung dokumentiert den Namen der aktuell bearbeiteten Projektdatei. Die Verbindungsdaten und Symbole der in dieser Datei vorhandenen Schaltpläne werden vom **Packager** in eine Layoutnetzliste mit Bauteilen und Netzen zusammengefasst.

[098] Die Datenbankstruktur ist beschaedigt (dateiname)!

Die DDB- bzw. DAT-Datei mit dem Namen **dateiname** hat nicht das erwartete Format. Dies kann auf einen Fehler auf dem Datenträger hinweisen oder rührt daher, dass ein vorangegangener Schreibvorgang in die Datei nicht korrekt beendet wurde. Beim Beginn des Schreibzugriffs in DDB- und DAT-Dateien wird im Header des Datei ein Flag gesetzt, das erst nach dem erfolgreichem Abschluss des Schreibvorgangs zuruückgesetzt wird. Das Vorhandensein dieses Flags wird ebenfalls mit dieser Meldung signalisiert.

Zur Problembehebung kann die -recover-Option des Utilityprogrammes **COPYDDB** verwendet werden. Sollte dies nicht zur Behebung des Problems führen kann die Projektdatei zur näheren Untersuchung an den Bartels-Support gesandt werden.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[097] Die Datenbankstruktur ist beschaedigt!

Eine nicht näher spezifizierte DDB- bzw. DAT-Datei hat nicht das erwartete Format. Dies kann auf einen Fehler auf dem Datenträger hinweisen oder rührt daher, dass ein vorangegangener Schreibvorgang in die Datei nicht korrekt beendet wurde. Beim Beginn des Schreibzugriffs in DDB- und DAT-Dateien wird im Header des Datei ein Flag gesetzt, das erst nach dem erfolgreichem Abschluss des Schreibvorgangs zuruückgesetzt wird. Das Vorhandensein dieses Flags wird ebenfalls mit dieser Meldung signalisiert.

Zur Problembehebung kann die -recover-Option des Utilityprogrammes **COPYDDB** verwendet werden. Sollte dies nicht zur Behebung des Problems führen kann die Projektdatei zur näheren Untersuchung an den Bartels-Support gesandt werden.

In Betracht kommen für diese Fehlermeldung die Projektdatei, die Standardbibliothek, die Alternativbibliothek oder eine der Setupdateien **bsetup.dat** (Setupdatei mit Bibliotheksnamen und Defaultlayoutelementnamen), **language.vdb** (alternative Ausgabemeldungen) oder **baewin.dat** (Fensterposition und -größe) im BAE-Programmverzeichnis (bzw. einem über die Umgebungsvariable **BAE_PROGDIR** gesetzen Alternativverzeichnis oder über die Umgebungsvariablen **BAE_BSETUP**, **BAE_LANG** oder **BAE_WINLIB** gesetzte Alternativsetupdateien).

[025] Einzel-Sub-Block 'blockname' mehrfach verwendet!

Der in den Schaltplaneinstellungen als Einzel-Sub-Block deklarierte hierarchische Schaltplanblock **blockname** wird von mehreren Blocksymbolen referenziert.

Zur Problembehebung sind überzählige Blockreferenzsymbole zu löschen oder der betreffende hierarchische Schaltplan ist als normaler mehrfach verwendbarer Sub-Block zu deklarieren.

[022] Es gibt keine Schaltplan Netzliste

In der für die Bearbeitung selektierten Projektdatei sind keine Schaltplannetzlisten hinterlegt. Sind in dem Projekt Schaltpläne ladbar deutet dies darauf hin, das diese lediglich als Gruppen gespeichert wurden, die keine logische Netzliste besitzen. In diesem Fall sind die Schaltpläne in den **Schaltplaneditor** zu laden und abzuspeichern. Beim Speichern wird eine logische Netzliste in der Projektdatei generiert. Beim Import von Netzlisten aus Fremdsystemen deutet diese Meldung darauf hin, dass das Einspielen der logischen Netzliste mit Hilfe der Anwenderfunktionen **CONCONV** oder **NETCONV** nicht erfolgreich war. Die Konvertierung sollte erneut gestartet und auf Fehlermeldungen geachtet werden.

[017] Es wurden keine Fehler festgestellt.

Gratulation, Sie haben soeben erfolgreich eine Layoutnetzliste für das angegebene Projekt generiert! Evtl. ausgegebenen Warnungsmeldungen sollte noch einmal Beachtung geschenkt werden, sie können Hinweise auf unbeabsichtigte unterschiedliche Attributzuweisungen für Symbole und dergleichen enthalten. Wird nach erfolgreichem Abschluss des **Packager**-Laufes unmittelbar in den **Layouteditor** gewechselt, so wird das zur erzeugten Layoutnetzliste gehörige gleichnamige Projektlayout automatisch in den **Layouteditor** geladen. Gibt es das Layout noch nicht, so wird das Anlegen des Layouts vorgeschlagen. Es ist zu beachten, das im **Layouteditor** in der Projektdatei zwar andersnamige Layouts angelegt werden können, diese haben aber keinen Bezug zu der erzeugten Layoutnetzliste wenn der Name des Layouts nicht identisch mit dem Namen der Layoutnetzliste ist.

[011] Fataler interner Fehler fehlernummer!

Beim **Packager**-Lauf ist ein unerwarteter Fehler aufgetreten (z.B. wenn die Suche nach einem Listenelement fehlschlägt, unmittelbar nachdem es in die Liste gespeichert wurde). Dies deutet entweder auf ein Problem mit der Rechnerhardware oder ein noch nicht bekanntes Problem im **Packager** hin. Tritt das Problem reproduzierbar nur bei einem Projekt auf, empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[054] Festnetz-Attribut 'attributname' nicht gefunden!

In der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols ist in einem **net**-Kommando das Attribut mit dem Namen **attributname** als Quelle für den Netznamen angegeben. Das Attribut ist an dem Symbol aber nicht gesetzt worden.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist das Attribut **attributname** an dem betreffenden Symbol zu setzen.

[055] Festnetz-Attribut 'attributname' ungueltig!

In der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols ist in einem **net**-Kommando das Attribut mit dem Namen **attributname** als Quelle für den Netznamen angegeben. Der Attributwert dieses Attributes stellt keinen gültigen Netznamen dar. In Netznamen unzulässige Zeichen sind **?**, *****, sowie nicht druckbare Zeichencodes. Gross-/Kleinschreibung spielt keine Rolle, die Namen werden automatisch in Kleinschreibung umgewandelt. Leerzeichen werden ausgefiltert und gelten somit nicht als ungültige Zeichen. Es wird dabei nur der erste nicht aus Leerzeichen bestehende Teilstring als Netznamen verwendet.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist in das Attribut **attributname** des betreffenden Symbols ein gültiger Netzname einzutragen.

[056] Festnetzdaten-Pin 'layoutbauteilpinname' nicht gefunden!

In der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols ist in einem **net**-Kommando dem Layoutbauteilpin mit dem Namen **layoutbauteilpinname** ein festes Netz für die Netzliste vorgegeben. Der Pin ist auf dem über die logische Definition des Symbols oder das Symbolattribut **$plname** angeforderten Layoutbauteilmakro aber nicht vorhanden.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es den Pin auf dem Layoutbauteilmakro gäbe und man kann diesen nachträglich auf dem Layoutbauteilmakro platzieren um ein zur Netzliste passendes Layout zu erhalten. Im Falle eines Fehlers wird bei einem unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem Symbol, das das Layoutbauteil referenziert, geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist das Attribut **attributname** an dem betreffenden Symbol zu setzen.

Zur Problembehebung ist der Pin auf dem Layoutbauteilmakro zu platzieren oder der Gehäusename in der logische Definition bzw. dem **$plname**-Attribut des Symbols zu korrigieren.

[113] Fragliches Bauteil : 'layoutbauteilname'

Bei der Bearbeitung des Layoutbauteils mit dem Namen **layoutbauteilname** ist ein Problem aufgetreten, das in den nachfolgenden Meldungszeilen noch genauer spezifiziert wird.

[122] Fragliches Layoutbauteil : 'layoutbauteilname' Pin 'layoutbauteilpinname' ('pinfunktionsname')

Die für den Pin **layoutbauteilpinname** des Layoutbauteils mit dem Namen **layoutbauteilname** definierte Pinfunktion **pinfunktionsname** erzeugt im angeschlossenen Netz einen Konflikt bzw. Fehler, der in den vorhergehenden Meldungszeilen genauer spezifiziert ist.

[117] Fragliches Mehrfachbauteil : 'layoutbauteilname' abweichender $noplc-Status :

In einem aus mehreren Symbolen/Gattern zusammengesetzten Layoutbauteil ist das **$noplc**-Attribut an den Untersymbolen unterschiedlich gesetzt. Der **Packager** gibt hierbei der Platzierungsanforderung Vorrang, wodurch aber auch als unplatziert markierte Untersymbole mitplatziert werden. Es bleibt dem Benutzer überlassen zu entscheiden, ob dies die Funktion der Schaltung beeinträchtigt. Die Symbole mit kollidierenden Platzierungsanforderungen werden in Folgemeldungszeilen aufgelistet.

Zur Problembehebung ist das **$noplc**-Attribut in den Untersymbolen ggf. auf einheitliche Werte zu setzen.

[114] Fragliches Symbol : 'symbolname'

Bei der Bearbeitung des Schaltplansymbols mit dem Namen **symbolname** ist ein Problem aufgetreten, das in den nachfolgenden Meldungszeilen noch genauer spezifiziert wird.

[115] Fragliches Variantenbauteil : 'layoutbauteilname'

Bei der Bearbeitung des Layoutbauteils mit dem Namen **layoutbauteilname** ist ein Problem aufgetreten, das in den nachfolgenden Meldungszeilen noch genauer spezifiziert wird.

[035] Fuer Variante variantennummer durch 'symbolname' ueberbelegt!

Das Symbol mit dem Namen **symbolname** möchte über sein **$vgrp**-Attribut in das in der vorhergehenden Meldungszeile dokumentierte Layoutvariantenbauteil gepackt werden und signalisiert durch sein in der Variante mit der Nummer **variantennummer** nicht gesetztes **$noplc**-Attribut, dass es in dieser Variante platziert werden soll. Das Layoutvariantenbauteil ist für diese Variante aber bereits durch ein anderes Symbol mit gleich gesetzten **$vgrp**-und **$noplc**-Attributen belegt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind die **$noplc**-Werte der in das betreffende über **$vgrp** selektierte Layoutvariantenbauteil gepackten Symbole so zu koordinieren, dass pro Variante nur jeweils ein Symbol für die Platzierung aktiv ist.

[101] Funktion fuer altes Format nicht verfuegbar!

Beim Zugriff auf eine mit einer sehr alten BAE-Version gespeicherte DDB- bzw. DAT-Datei ist ein Problem aufgetreten. Da der **Packager** für alle betreffenden alten Formateinträge Behandlungsroutinen implementiert sind, deutet diese Meldung auf ein internes Problem im **Packager** hin, und die Projektdatei sollte zur näheren Untersuchung an den Bartels-Support gesendet werden.

[071] Kein Schaltplanblatt mit Blockname 'blockname' gefunden!

In den logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Blocksymbols eines hierarchischen Schaltplans wurde über das **call**-Kommando ein **blockname** referenziert, der auf keinem der als Sub-Block bzw. Einzel-Sub-Blöcke deklarierten Schaltpläne des Projektes als **Plan Blockname** definiert ist, d.h. der oder die Schaltpläne zu dem Blocksymbol fehlen.

Zur Problembehebung sind die Schaltpläne zu zeichnen, bzw. die Blocknamen im **call**-Kommando oder den Planeinstellungen zu korrigieren.

[037] Keine Zuweisung fuer Symbolpin 'symbolpinname' gefunden!

Der auf dem in der vorhergehenden Meldungszeile aufgelisteten Symbol vorhandene Pin mit dem Namen **symbolpinname** ist in der logische Definition für das Symbol im **pin**-Kommando nicht aufgelistet bzw. falls kein **pin**-Kommando in der logischen Definition vorhanden ist war keine 1:1-Namenszuordnung zu einem Layoutbauteilpin des verwendeten Layoutbauteilmakros möglich, da es auf dem Layoutbauteilmakro keinen Pin mit diesem Namen gibt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der fragliche Pin mit einem **xlat**-Kommando der logischen Definition einem existierenden Layoutbauteilpin zuzuordnen oder in das **pin**-Kommando der logischen Definition aufzunehmen oder auf dem Layoutbauteilmakro zu platzieren oder auf dem Schaltplansymbol in einen in den **pin**-Kommandos definierten Namen umzubenennen. Es ist zu beachten, dass nach dem Platzieren/Löschen/Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen/geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden die Daten aber erst beim Speichern der Schaltplanblätter übertragen.

[009] Layout Elementname ....: 'layoutnetzlistenname'

Diese Meldung dokumentiert den Namen der vom Packger zu erzeugenden Layoutnetzliste. Dieser sollte identisch mit dem später für das Layout verwendeten Elementnamen sein, das der Bezug zwischen Layout und Layoutnetzliste über den Namen hergestellt wird. Wird das Layout anders benannt, so werden die Einträge der Netzliste nicht berücksichtigt und eine leere Netzliste mit dem Namen des Layoutelementes erzeugt und verwendet.

[042] Layoutbauteilmakro 'layoutbauteilmakroname' nicht in Bibliothek!

Das von dem in der vorhergehenden Meldungszeile aufgelisteten Symbol über die logische Definition oder das Attribut **$plname** angeforderte Layoutbauteilmakro mit dem Namen **layoutbauteilmakroname** konnte weder in der Projektdatei noch in den angegebenen Bibliotheken gefunden werden.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es das Layoutbauteilmakro gäbe (für die referenzierten Pins wird dann noch jeweils eine eigene Warnung ausgegeben) und man kann das Layoutbauteilmakro erst nachträglich erstellen um ein zur Netzliste passendes Layout zu erhalten. Im Falle eines Fehlers wird bei einem unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist das Layoutbauteilmakro zu erstellen oder der Gehäusename in der logische Definition bzw. dem **$plname**-Attribut des Symbols zu korrigieren (Gross-/Kleinschreibung spielt im **$plname**-Attribut keine Rolle).

[043] Logische Definition zu 'definitionsname' nicht in Bibliothek!

Die von dem in der vorhergehenden Meldungszeile aufgelisteten Symbol angeforderte logische Definition mit dem Namen **definitionsname** konnte weder in der Projektdatei noch in den angegebenen Bibliotheken gefunden werden. Der Name der logischen Definition ist per Default der Name des Symbolmakros. Über das Attribut **$rlname** (requested logical library name) kann auch ein alternativer vom Symbolmakronamen abweichender Definitionsname selektiert werden. Mit dem Attribut **$rlext** kann zur Definitionsnamensbildung dem Symbolmakronamen bzw. dem Wert von **$rlname** eine Namenserweiterung der Art **basisname_rlextwert** angehängt werden. Bei über das **$vgrp**-Attribut zusammengepackten variantenabhängig belegten Layoutbauteilen können **$rlname**-und **$rlext**-Attribute auch von den mit gleichem **$vgrp** benannten Nachbarsymbolen verwendet werden, wenn das Symbol keine eigenen Zuweisungen für diese Attribute besitzt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Behebung des Problems kann entweder die angeforderte logische Definition angelegt werden, in die verwendeten Bibliotheken/das Projekt kopiert werden, andere Bibliotheken verwendet werden oder die angeforderte logische Definition in den Attributen **$rlname** bzw. **$rlext** korrigiert werden.

[040] Logischer Pin 'symbolpinname' kann nicht umgesetzt werden!

Der auf dem in der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols im **pin**-Kommando aufgelistete Pin mit dem Namen **symbolpinname** ist in keinem der **xlat**-Kommandos mit Zuordnungen zu Layoutbauteilpins aufgeführt und es ist auch keine 1:1-Namenszuordnung zu einem Layoutbauteilpin des verwendeten Layoutbauteilmakros möglich, da es auf dem Layoutbauteilmakro keinen Pin mit diesem Namen gibt.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es den Pin auf dem Layoutbauteilmakro gäbe und führt eine 1:1-Zuweisung des Symbolpins zu diesem Pin durch und man kann diesen nachträglich auf dem Layoutbauteilmakro platzieren um ein zur Netzliste passendes Layout zu erhalten. Im Falle eines Fehlers wird bei einem unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der fragliche Pin in die **xlat**-Kommandos der logischen Definition aufzunehmen oder auf dem Layoutbauteilmakro zu platzieren oder auf dem Schaltplansymbol in einen in den **xlat**-Kommandos definierten Namen umzubenennen. Es ist zu beachten, dass nach dem Platzieren/Löschen/Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen/geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden die Daten aber erst beim Speichern der Schaltplanblätter übertragen.

[126] Netz 'netzname' $nettype=mixed da 'netztyp1'<>'netztyp2' von 'symbolname'!

Dem Netz mit dem Namen **netzname** wurden über das Netzattribut **$nettype** von verschiedenen Symbolen aus unterschiedliche Netztypen zugewiesen. **netztyp1** gibt dabei den zuerst gefundenen Netztyp an und **netztyp2** den davon abweichenden zweiten Netztyp, der über das Symbol mit dem Namen **symbolname** zugewiesen wurde. Wird für den Symbolnamen ein Leerstring ausgegeben, stammt die Netztypzuweisung aus einer dem Netzlabel zugeordneten logischen Definition. Bei Netzen mit unterschiedlichen Netztypzuweisungen trägt der **Packager** automatisch den Netztyp **mixed** ein.

Zur Problembehebung sind die Netztypzuweisungen für ein Netz zu vereinheitlichen.

[030] Netz 'netzname' hat mehrere Netznummern!

Das Netz mit dem Namen **netzname** ist in der logischen Netzliste des unter "Aktiver Planname" dokumentierten Schaltplanblattes mit unterschiedliche Netznummern aufgeführt. Dies ist nicht erlaubt. Da die Netznummern im BAE automatisch vergeben werden, deutet dies evtl. auf ein internes Problem hin.

Zur Problembehebung kann versucht werden das betreffende Schaltplanblatt im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei wird die logische Netzliste neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

Wurde die logische Netzliste mit Hilfe der Anwenderfunktionen **CONCONV** oder **NETCONV** aus einem Fremdsystem importiert, ist zu überprüfen, ob die ungültige Mehrfachzuweisung von Netznummern nicht von entsprechenden Vorgaben in der Quellnetzliste herrührt.

[086] Netz 'netzname' hat nur Eingaenge!

An das Netz mit dem Namen **netzname** sind ein oder mehrere Pins angeschlossen, die über das Attribut **$pintype** als vom Typ **in** (d.h. Eingang) deklariert sind aber keine Pins vom Typ **out** (output, d.h. Ausgang), **bidi** (bidirektional) oder **sup** (supply, d.h. Versorgungsspannung). Das Netz befindet sich dadurch auf keinem definierten Potential und die Funktionsfähigkeit der Schaltung ist zweifelhaft. Die Pins des Netzes und ihre Funktionen werden in folgenden Meldungszeilen aufgelistet.

Zur Problembehebung ist das Netz durch entsprechende Pins auf definiertes Potential zu legen.

[031] Netz 'netzname' hat ungueltige negative DRC-Blockzuweisung!

Es wurde versucht dem Netz mit dem Namen **netzname** über das **$drcblk**-Netzattribut (das in der logischen Definition auch durch einen anderen Symbolattributnamen ersetzt oder auf einen festen Wert gesetzt werden kann) des in der vorhergehenden Meldungszeile aufgelisteten Symbols eine negative DRC-Blocknummer zuzuweisen. Es sind nur positive DRC-Blocknummern erlaubt größer gleich Null erlaubt. In den Standardbibliotheken des BAE kommen nur die Symbole **att_drcblk**, **tag_net_drcblk**, **tag_netpin_drcblk** und **tag_netarea_drcblk** aus der Bibliothek **route.ddb** für diese Fehlermeldung in Betracht.

Die DRC-Blocknummer wird nur in der **HighEnd**-Version des **Layouteditors** für den Abstands-DRC berücksichtigt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem die ungütige Zuweisung vornehmenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol in der logischen Definition bzw. im Schaltplan ein gültiger Wert für die DRC-Blocknummer des Netzes zuzuweisen.

[029] Netz 'netzname' hat ungueltige negative Nummer!

Das Netz mit dem Namen **netzname** ist in der logischen Netzliste des unter "Aktiver Planname" dokumentierten Schaltplanblattes mit einer negativen Netznummer aufgeführt. Dies ist nicht erlaubt. Da die Netznummern im BAE automatisch vergeben werden, deutet dies evtl. auf ein internes Problem hin.

Zur Problembehebung kann versucht werden das betreffende Schaltplanblatt im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei wird die logische Netzliste neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

Wurde die logische Netzliste mit Hilfe der Anwenderfunktionen **CONCONV** oder **NETCONV** aus einem Fremdsystem importiert, ist zu überprüfen, ob die ungültige Netznummer nicht von einer entsprechenden Vorgabe in der Quellnetzliste herrührt.

[032] Netz 'netzname' sind verschiedene DRC-Blocknummern

zugewiesen!

Es wurde versucht dem Netz mit dem Namen **netzname** über mehrere **$drcblk**-Netzattribute unterschiedliche DRC-Blocknummern zuzuweisen. Da nur eine DRC-Blocknummer pro Netz erlaubt ist, ist die Zuordnung somit nicht eindeutig.

Die DRC-Blocknummer wird nur in der **HighEnd**-Version des **Layouteditor** für den Abstands-DRC berücksichtigt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem Symbol, das die unterschiedliche Zuweisung vornimmt geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist eine der DRC-Blocknummernzuweisungen zurückzunehmen.

[120] Netz 'netzname' verschiedene Werte 'attributname' ('wert1'<>'wert2' von 'symbolname')!

Dem Netz mit dem Namen **netzname** sind für das Netzattribut **attributname** unterschiedliche Werte zugewiesen worden. Dies kann durch in den logischen Definition von an das Netz angeschlossenen Labels und Symbolen vorgenommene Netzattributzuweisungen und/oder über Netzattributzuweisungen mit an das Netz zugewiesenen Tagsymbolen geschehen sein. Der Name des den Konflikt verursachenden Symbols wird unter **symbolname** angegeben.

Zur Problembehebung sind die Netzattributzuweisungen in den logischen Definitionen oder den Tagsymbolen anzugleichen oder überzählige Attributzuweisungen zu entfernen.

Einen Sonderfall stellt das Netzpinattribut **$net** dar, das von **Packager** und Backannotation automatisch auf den Namen des angeschlossenen Netzes gesetzt wird. Beim ersten Packagerlauf nach Netzlistenänderungen werden diese Warnungsmeldungen durch Kollision alter Pinnetznamen mit neuen Pinnetznamen verursacht und können ignoriert bzw. als Dokumentation der Netznamensänderungen (vor allem bei den automatisch generierten mit **@** beginnenden Netznamen) verwendet werden. Nach dem Abgleich der **$net**-Attribute durch den **Packager** treten die Warnungsmeldungen bei folgenden **Packager**-Läufen nicht mehr auf.

[023] Netzliste 'netlistenname' nicht gefunden!

Die logische Netzliste mit dem Namen **netlistenname** konnte in der Projektdatei nicht gefunden werden. Da die Netzlisten anhand einer zu Beginn des **Packager**-Laufes durch Zugriff auf die Projektdatei gebildeten Liste geladen werden, deutet dies auf eine Änderung der Projektdatei während des **Packager**-Laufes hin (z.B. durch einen anderen Anwender im Netzwerk).

Zur Problembehebung kann versucht werden, das zu der logischen Netzliste gehörige gleichnamige Schaltplanblatt im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei wird die logische Netzliste neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[013] Netzlistenaenderung in Layout 'layoutname'/'projektdateiname'! Packagerlauf trotzdem durchfuehren?

In der Projekdatei **projektdateiname** wurden im Layout mit dem Namen **layoutname** Änderungen durchgeführt, die eine Backannotation erforderlich machen. Diese Backannotation wurde noch nicht durchgeführt. Bei Bestätigung der Abfrage werden die im Layout durchgeführten Netzlistenänderungen wie Bauteilumbenennungen, Alternativbauformwechsel und Pin/-Gate-Swaps verworfen. Die Bauteilplatzierungen und Leiterbahnverbindungen des Layouts passen dann eventuell nicht mehr zur erzeugten Netzliste.

Es ist im Allgemeinen zu empfehlen, zunächst in den **Schaltplaneditor** zu wechseln und dort die Backannotation durchzuführen. Sie wird beim Modulwechsel automatisch vorgeschlagen, kann alternativ aber auch mit Utilities / Backannotation von Hand durchgeführt werden.

[125] Netzname zu 'layoutbauteilname'/'layoutbauteilpinname' zu lang!

Bei der automatischen Netznamensgenerierung über eine unter Unbenannte Netze aktivierte Option wurde ein Netzname gebildet, der die Maximallänge von 40 Zeichen überschreitet.

Zur Problembehebung ist der Name des Layoutbauteils oder der Netznamensprefix entsprechend zu kürzen.

[124] Netznamenskollision 'netzname'!

Bei der automatischen Netznamensgenerierung über eine unter Unbenannte Netze aktivierte Option wurde aus Netznamensprefix, Layoutbauteilnamen und Layoutbauteilpinnanem ein Netzname gebildet, der im Schaltplan bereits explizit vergeben wurde. Durch Zusammenfassung der beiden Netze über den gemeinsamen Namen würden ungewollte Verbindungen und damit Kurzschlüsse entstehen.

Zur Problembehebung ist das Konfliktnetz im Schaltplan, das Layoutbauteil oder der Netznamensprefix umzubenennen.

[002] Nicht implementiert!

Ein aufgerufener Menüpunkt konnte nicht gestartet werden. Dies deutet auf eine installierte Zwischenversion hin, in der neue Menüpunkte auf Vorrat definiert, aber noch nicht implementiert wurden. Die Funktion/Option wird in einer der nächsten Versionen verfügbar sein.

[021] Optimierung Ende (n Bauteile entfernt).

Diese Meldung signalisiert das Ende des Optimierungslaufes und listet die Anzahl der aus der Netzliste entfernten Bauteile. Beim Optimieren werden Bauteile aus der Netzliste entfernt, die über den Wert `yes` für das Attribut `$optimize` für die Optimierung freigegeben sind und bei denen sämtliche über das Attribut `$pintype` als `out` (output d.h. Ausgang) deklarierten Pins keine weiterführenden Verbindungen besitzen. Dies findet im Normalfall nur für über das `architecture`-Kommando in der logischen Definition synthetisierte Logikbausteine Anwendung.

[020] Optimierung Start (n Bauteile vorhanden).

Diese Meldung signalisiert den Start des Optimierungslaufes und listet die Anzahl der ursprünglich vorhandenen Netzlistenbauteile. Beim Optimieren werden Bauteile aus der Netzliste entfernt, die über den Wert `yes` für das Attribut `$optimize` für die Optimierung freigegeben sind und bei denen sämtliche über das Attribut `$pintype` als `out` (output d.h. Ausgang) deklarierten Pins keine weiterführenden Verbindungen besitzen. Dies findet im Normalfall nur für über das `architecture`-Kommando in der logischen Definition synthetisierte Logikbausteine Anwendung.

[083] Pin 'layoutbauteilname/layoutbauteilpinname' verschiedene Werte fuer 'attributname' ('wert1'<>'wert2')!

Dem Pin `layoutbauteilpinname` des Layoutbauteils `layoutbauteilname` wurden für das Attribut `attributname` die beiden unterschiedlichen Attributwerte `wert1` und `wert2` zugewiesen. Als Quelle für die unterschiedlichen Attributwerte des Layoutbauteilpins kommen ein `newattr`-Kommando in der logischen Definition des zugehörigen Symbols oder an den zugehörigen Symbolpin angehängte Tagpinsymbole mit Attributzuweisungen in Betracht. Dies ist nur eine Warnungsmeldung, der **Packager** setzt die Bearbeitung des Projektes fort und verwendet `wert1` als Pinattributwert, der `wert2` wird verworfen.

Zur Behebung des Problems sind die Attributzuweisungen der Tagpinsymbole bzw. der logischen Definitionen anzugleichen.

[085] Pin 'layoutbauteilname/layoutbauteilpinname' verschiedene Werte fuer 'attributname'/variantennummer('wert1'<>'wert2')!

Dem Pin **`layoutbauteilpinname`** des Layoutbauteils **`layoutbauteilname`** wurden in der Variante mit der Nummer **`variantennummer`** für das Attribut **`attributname`** die beiden unterschiedlichen Attributwerte **`wert1`** und **`wert2`** zugewiesen. Als Quelle für die unterschiedlichen Attributwerte des Layoutbauteilpins kommen ein **`newattr`**-Kommando in der logischen Definition des zugehörigen Symbols oder an den zugehörigen Symbolpin angehängte Tagpinsymbole mit Attributzuweisungen in Betracht. Dies ist nur eine Warnungsmeldung, der **Packager** setzt die Bearbeitung des Projektes fort und verwendet **`wert1`** als Pinattributwert, der **`wert2`** wird für diese Variante verworfen.

Zur Behebung des Problems sind die Attributzuweisungen der Tagpinsymbole bzw. der logischen Definitionen für die betreffende Variante anzugleichen.

[075] Pin 'layoutbauteilpinname' fuer Attribut 'attributname' nicht gefunden!

Der von einem **`newattr`**-Kommando in der logischen Definition des in der vorhergehenden Meldungszeile dokumentierten Symbols oder Netzlisteeinträge für das Pinattribut **`attributname`** referenzierte Layoutbauteilpin mit dem Namen **`layoutbauteilpinname`** ist auf dem über die logische Definition des Symbols oder das Symbolattribut **`$plname`** angeforderten Layoutbauteilmakro nicht vorhanden.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der Pin auf dem Layoutbauteilmakro zu platzieren oder der Pin- bzw. Gehäusename in der logische Definition bzw. dem **`$plname`**-Attribut des Symbols zu korrigieren.

[069] Pin 'layoutbauteilpinname' hat ungueltige Pin-Funktion 'pinfunktionsspezifikation'!

Dem Pin mit dem Namen **`layoutbauteilpinname`** (der Layoutbauteilname ist in der vorhergehenden Meldungszeile dokumentiert) wird über das Pinattribut **`$pintype`** die nicht unterstützte Pinfunktion **`pinfunktionsspezifikation`** zugewiesen. Erlaubt sind nur die Pinfunktionen **`in`** (Input/Eingang), **`out`** (Output/Ausgang), **`bidi`** (bidirektional), **`anl`** (Analog) und **`sup`** (supply/Versorgung) wobei Gross-/Kleinschreibung keine Rolle spielt. Stammt die Zuweisung nicht aus einer logischen Definition sondern aus einen Tagpinsymbol, so wird statt des **`layoutbauteilpinname`** (und Layoutbauteilname in der vorhergehenden Meldungszeile) der Name des Tagsymbolpins und Tagsymbols angezeigt. (in den Standardbibliotheken des BAE kommt hierbei nur das Symbol **`tag_pin_pintype`** aus der Bibliothek **`route.ddb`** in Betracht).

Dies ist nur eine Warnungsmeldung, die nicht zum Abbruch des **Packager**-Laufes führt. Der Pin wird im weiteren Verlauf so behandelt, als ob er keine **`$pintype`**-Zuweisung besitzt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem zugehörigen Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist die **`$pintype`**-Zuweisung in der logischen Definition bzw. im referenzierenden Tagpinsymbol auf einen gültigen Wert aus obiger Liste zu bringen.

[070] Pin 'layoutbauteilpinname' hat widersprechende Pin-Funktionen!

Dem Pin mit dem Namen **layoutbauteilpinname** (der Layoutbauteilname ist in der vorhergehenden Meldungszeile dokumentiert) wurden über das Pinattribut **$pintype** unterschiedliche Pinfunktionen zugewiesen. Dies ist nicht erlaubt. Die unterschiedlichen Zuweisungen können durch Tagpinsymbole (in den Standardbibliotheken des BAE das Symbol **tag_pin_pintype** aus der Bibliothek **route.ddb**) und **newattr "$pintype"**-Kommandos in der logischen Definition des Symboles vorgenommen worden sein.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem Symbol, das den zugeordneten Pin enthält geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind die **$pintype**-Zuweisungen in der logischen Definition bzw. den referenzierenden Tagpinsymbolen auf einen einheitlichen Wert zu bringen, bzw. überzählige Tagpinsymbole zu entfernen.

[053] Pin 'layoutbauteilpinname' ist mehreren Teilen zugeordnet!

Das in der vorhergehenden Meldungszeile aufgelistete Symbol referenziert über seine logische Definition oder eine 1:1-Zuordnung der Symbolpins zu Layoutbauteilpins einen Layoutbauteilpin mit dem Namen **layoutbauteilpinname**. Dieser wird bereits von einem anderen in das Layoutbauteil gepackten Symbol mit einer anderen logischen Definition referenziert. Die Mehrfachreferenzierung des Pins stammt vermutlich aus den unterschiedlichen logischen Definitionen einer **mainpart**/**subpart**-Kombination. Dabei ist zu bedenken, dass ein **mainpart** ohne Pins automatisch eine Komplettbelegung des Layoutbauteils mit Dummynetzen auslösst. Bei einem **mainpart** ohne Pins kommt es also unweigerlich zu dieser Fehlermeldung, sobald ein **subpart** mit Pins referenziert wird.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind die Symbolpinzuordnungen für den Layoutbauteilpin in den logischen Definitionen zu korrigieren, bzw. das **mainpart** so zu definieren, dass es mindestens einen Pin mit Zuordnung besitzt.

[038] Pin 'layoutbauteilpinname' mehrfach/verschieden verwendet!

Das in der vorhergehenden Meldungszeile aufgelistete Symbol referenziert über seine logische Definition oder eine 1:1-Zuordnung der Symbolpins zu Layoutbauteilpins einen Layoutbauteilpin mit dem Namen **layoutbauteilpinname**. Dieser wird bereits von einem anderen in das Layoutbauteil gepackten Symbol referenziert und ist dort an ein anderes Netz angeschlossen. Da beim Einspielen der logischen Definition für ein Symbol ein Konsistenzcheck für die verwendeten Pins durchgeführt wird stammt Mehrfachreferenzierung des Pins mit hoher Wahrscheinlichkeit aus den unterschiedlichen logischen Definitionen einer **mainpart**/**subpart**-Kombination.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind die Symbolpinzuordnungen für den Layoutbauteilpin in den logischen Definitionen zu korrigieren.

[073] Pin 'symbolbuspinname.anzapfungsname' Bus Name ist zu lang!

Der Symbolbuspin mit dem Namen **symbolbuspinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) referenziert über ein **xlat**-Kommando in der logischen Definition des Symbols eine Busanzapfung mit dem Namen **anzapfungsname**. Bei der internen Netznamensbildung für das Netz der Anzapfung wurde die maximale Netznamenslänge von 40 Zeichen überschritten. Der interne Netznamen wird nach dem Schema **busname.anzapfungsname** gebildet.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem den Buspin enthaltenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der Busname im Schaltplan durch Umbenennen des Netzlabels entsprechend zu kürzen oder ein kürzerer Name für die Busanzapfung zu wählen.

[068] Pin 'symbolpin' hat ungueltige Netz-Prioritaet!

Dem Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition über ein **netattr priority**-Kommando ein Attributname oder fester Wert für die Bearbeitungspriorität des am **symbolpinname** angeschlossenen Netzes zugewiesen. Dieser feste Wert oder der Wert des angegegbenen Attributes ist negativ. Negative Prioritätswerte sind nicht erlaubt. In den Standardbibliotheken des BAE kommen nur die Symbole **att_pr**, **tag_net_priority**, **tag_netpin_priority** und **tag_netarea_priority** aus der Bibliothek **route.ddb** für diese Fehlermeldung in Betracht.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol in der logischen Definition bzw. im Schaltplan ein positiver Wert größer gleich Null für die Bearbeitungspriorität des Netzes im Autorouter zuzuweisen.

[064] Pin 'symbolpinname' Netz-Attribut fehlt!

Dem Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition des Symbols über ein **netattr**-Kommando ein Attributname für ein Netzattribut zugeordnet, das Attribut wurde am betreffenden Symbol aber nicht gesetzt. In den Standardbibliotheken des BAE kommen nur Symbole aus der Bibliothek**route.ddb** für diese Fehlermeldung in Betracht.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol im Schaltplan ein Wert für das Netzattribut zuzuweisen (ansonsten wäre das Symbol überflüssig und könnte genausogut gelöscht werden).

[041] Pin 'symbolpinname' Netznummer ist ungueltig!

Dem Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) ist in der logischen Netzliste des unter "Aktiver Planname" dokumentierten Schaltplanblattes eine Netznummer zugeordnet, zu der es in der Netzliste keine entsprechnung gibt. Da die Netznummern im BAE automatisch vergeben werden, deutet dies evtl. auf ein internes Problem hin.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung kann versucht werden das betreffende Schaltplanblatt im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei wird die logische Netzliste neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

Wurde die logische Netzliste mit Hilfe der Anwenderfunktionen **CONCONV** oder **NETCONV** aus einem Fremdsystem importiert, ist zu überprüfen, ob die ungültige Netznummerzuweisung nicht von entsprechenden Vorgaben in der Quellnetzliste herrührt.

[074] Pin 'symbolpinname' fuer Gate Vorgabe nicht gefunden!

In der Layoutnetzliste des Projektes sind Pin-/Gateswapinformationen für den Symbolpin mit dem Namen `symbolpinname` (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) hinterlegt, der Symbolpin wurde in der logischen Netzliste des Schaltplanes aber nicht gefunden, d.h. er ist auf Schaltplansymbol nicht mehr vorhanden. Es ist nicht erlaubt, einmal im Layout geswappte Pins vom zugehörigen Schaltplansymbol zu entfernen (oder umzubenennen).

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der Pin wieder auf dem Symbol zu platzieren. Es ist zu beachten, dass nach dem Platzieren/Löschen/Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen/geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden die Daten aber erst beim Speichern der Schaltplanblätter übertragen.

Ist es wirklich zwingend erforderlich einen Symbolpin nach dem Swap im Layout zu löschen/umzubenennen, muss ein neuer Symbolname/LAyoutbauteilname vergeben werden. Die Swapinformationen eines namentlich komplett aus der Netzliste entfernten Layoutbauteils werden verworfen. Mit Bauteile / Namen aendern kann das alte Layoutbauteil an gleicher Position umbenannt werden. Die Swaps müssen von Hand nachgetragen werden.

[066] Pin 'symbolpinname' hat ungueltige Routing-Breite!

Dem Symbolpin mit dem Namen `symbolpinname` (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition über ein `netattr routwidth`-Kommando ein Attributname oder fester Wert für die Standardleiterbahnbreite des am `symbolpinname` angeschlossenen Netzes zugewiesen. Dieser feste Wert oder der Wert des angegegbenen Attributes repräsentiert keine gültige Bahnbreite. Die Bahnbreite ist in Millimetern mit einem `.` als Dezimaltrennzeichen anzugeben und muss ein Wert größer als Null sein. Die Umwandlung der Zahl wird beim ersten ungültigen Zeichen beendet. Es ist also z.B. erlaubt zur Dokumentation ein `mm` an das Ende des Attributwertes zu setzen, führende Kommentare sind jedoch nicht erlaubt. In den Standardbibliotheken des BAE kommen nur die Symbole `att_rw`, `tag_net_routwidth`, `tag_netpin_routwidth` und `tag_netarea_routwidth` aus der Bibliothek `route.ddb` für diese Fehlermeldung in Betracht.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem die ungültige Zuweisung vornehmenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol in der logischen Definition bzw. im Schaltplan ein gültiger Wert für die Standardleiterbahnbreite des Netzes zuzuweisen.

[065] Pin 'symbolpinname' hat ungueltige Versorgungs-Breite!

Dem Symbolpin mit dem Namen `symbolpinname` (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition über ein `netattr powwidth`-Kommando ein Attributname oder fester Wert für die Anschlussbreite von Versorgunspins an dem am `symbolpinname` angeschlossenen Netz zugewiesen. Dieser feste Wert oder der Wert des angegegbenen Attributes repräsentiert keine gültige Anschlussbreite. Die Anschlussbreite ist in Millimetern mit einem `.` als Dezimaltrennzeichen anzugeben und muss ein Wert größer als Null sein. Die Umwandlung der Zahl wird beim ersten ungültigen Zeichen beendet. Es ist also z.B. erlaubt zur Dokumentation ein `mm` an das Ende des Attributwertes zu setzen, führende Kommentare sind jedoch nicht erlaubt. In den Standardbibliotheken des BAE kommen nur die Symbole `att_pw`, `tag_net_powwidth`, `tag_netpin_powwidth` und `tag_netarea_powwidth` aus der Bibliothek `route.ddb` für diese Fehlermeldung in Betracht.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem die ungültige Zuweisung vornehmenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol in der logischen Definition bzw. im Schaltplan ein gültiger Wert für die Versorgungsanschlussbreite des Netzes zuzuweisen.

[067] Pin 'symbolpinname' hat ungueltigen Min.-Abstand!

Dem Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition über ein **netattr mindist**-Kommando ein Attributname oder fester Wert für den Mindestabstand des am **symbolpinname** angeschlossenen Netzes zugewiesen. Dieser feste Wert oder der Wert des angegegbenen Attributes repräsentiert keinen gültigen Mindestabstand. Der Mindestabstand ist in Millimetern mit einem **.** als Dezimaltrennzeichen anzugeben und muss ein Wert größer als Null sein. Die Umwandlung der Zahl wird beim ersten ungültigen Zeichen beendet. Es ist also z.B. erlaubt zur Dokumentation ein **mm** an das Ende des Attributwertes zu setzen, führende Kommentare sind jedoch nicht erlaubt. In den Standardbibliotheken des BAE kommen nur die Symbole **att_md**, **tag_net_mindist**, **tag_netpin_mindist** und **tag_netarea_mindist** aus der Bibliothek **route.ddb** für diese Fehlermeldung in Betracht.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem die ungültige Zuweisung vornehmenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist dem Symbol in der logischen Definition bzw. im Schaltplan ein gültiger Wert für den Mindestabstand des Netzes zuzuweisen.

[039] Pin 'symbolpinname' verschieden gepolt verwendet!

Das in der vorhergehenden Meldungszeile aufgelistete Symbol ist über ein **$vgrp**-Attribut zusammen mit anderen Symbolen mit gleichem **$vgrp**-Attributwert in ein variantenabhängig belegtes Layoutbauteil eingebettet. Der Pin mit dem Namen **symbolpinname** ist bei diesem Symbol an ein anderes Netz angeschlossen, als die korrespondierenden Pins der anderen Symbole. Dies ist nicht erlaubt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind alle Symbole des Variantenbauteiles im Schaltplan parallel zu schalten.

[072] Port-Pin 'layoputbauteilpinname' nicht gefunden!

Der im **architecture**-Kommando der logischen Definition des in der vorhergehenden Meldungszeile aufgeführten Symbols aufgelistete Laoyutbauteilpin mit dem Namen **layoutbauteilpinname** wurde nicht gefunden.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es den Pin auf dem Layoutbauteilmakro gäbe und man kann diesen nachträglich auf dem Layoutbauteilmakro platzieren um ein zur Netzliste passendes Layout zu erhalten.

Zur Behebung des Problems ist der Name des Pins im **architecture**-Kommando zu korrigieren oder der entsprechende Pin auf dem Layoutbauteil zu platzieren.

Diese Meldung könnte auch bei der Bearbeitung von Blocksymbolen für hierarchische Schaltplanblätter auftreten, deutet hier aber auf einen noch nicht bekannten Fehler im **Packager** hin, da Port-Pins von Blocksymbolen vor der Verwendung automatisch definiert sein sollten. Es kann versucht werden die Schaltplanblätter des Projektes im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei werden die logischen Netzlisten neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[015] Schreibfehler, die Datenbank ist gestoert!

Diese Fehlermeldung wird ausgegeben, wenn in einem der **Packager**-Schritte die Schreibzugriffe auf die Projektdatei durchführen ein Fehler aufgetreten ist. Dies muss nicht notwendigerweise ein Schreibfehler gewesen sein der eine gestörte Datenbankstruktur hinterlassen hat (siehe vor dieser Meldung ausgegebene Fehlermeldungen). Es ist allerdings nicht davon auszugehen, dass eine korrekte Layoutnetzliste geschrieben wurde.

Zur Problembehebung siehe die vorangehenden Fehlermeldungen. Falls die Projektdatei tatsächlich unlesbar geworden ist, kann die -recover-Option des Utilityprogrammes **COPYDDB** zur Restauration der Projektdatei verwendet werden. Sollte dies nicht zur Behebung des Problems führen, dann kann die Projektdatei zur näheren Untersuchung an den Bartels-Support gesandt werden.

[057] Swapdaten-Pin 'layoutbauteilpinname' nicht gefunden!

In der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols ist in einem **`xlat`**-Kommando ein Layoutbauteilpin mit dem Namen **`layoutbauteilpinname`** aufgeführt. Dieser Pin ist auf dem über die logische Definition des Symbols oder das Symbolattribut **`$plname`** angeforderten Layoutbauteilmakro aber nicht vorhanden.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es den Pin auf dem Layoutbauteilmakro gäbe und man kann diesen nachträglich auf dem Layoutbauteilmakro platzieren um ein zur Netzliste passendes Layout zu erhalten. Im Falle eines Fehlers wird bei einem unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem Symbol, das das Layoutbauteil referenziert, geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der Pin auf dem Layoutbauteilmakro zu platzieren oder der Pin- bzw. Gehäusename in der logische Definition bzw. dem **`$plname`**-Attribut des Symbols zu korrigieren.

[137] Symbol 'symbolname' 'attributname' nicht eindeutig ('wert1'<>'wert2')!

Dem Symbol **`symbolname`** wurden in der logischen Definition über **`newattr`**-Kommandos mehrere unterschiedliche Werte für das Attribut **`attributname`** zugewiesen.

Zur Problembehebung sind die überzähligen **`newattr`**-Kommandos aus der logischen Definition für das Symbol zu entfernen.

[077] Symbol 'symbolname' Definitionszuweisung hat ungueltige Klasse: ('symbolmakroklasse'<>'definitionsklasse')!

Die für das Symbol mit dem Namen **`symbolname`** mit Hilfe von **`$rlname`**-bzw. **`$rlext`**-Attributeinträgen selektierte logische Definition besitzt eine andere Klasse als die zum Symbolmakronamen gehörige Definition.

Die Klasse wird in der logischen Definition über ein **`class`**-Kommando angegeben und dient dazu eine Konsistenzprüfung der über **`$rlname`**-und **`$rlext`**-Attribute selektierten Definitionen durchzuführen, so dass die versehentliche Zuweisung einer nicht zum Symbolmakro passenden logischen Definition vermieden wird.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem Symbol, das die ungültige Zuweisung vornimmt geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[079] Symbol 'symbolname' Definitionszuweisung nicht erlaubt!

Es wurde versucht für das Symbol mit dem Namen **symbolname** mit Hilfe von **$rlname**-bzw. **$rlext**-Attributeinträgen eine alternative logische Definition zu selektieren. In der zum Symbolmakronamen gehörigen Definition ist aber kein **class**-Kommando angegeben.

Mit dem **class**-Kommando wird die Klasse einer logischen Definition angegeben, die dazu dient eine Konsistenzprüfung der über **$rlname**-und **$rlext**-Attribute selektierten Definitionen durchzuführen, so dass die versehentliche Zuweisung einer nicht zum Symbolmakro passenden logischen Definition vermieden wird. Das **class**-Kommando ist demnach für Symbolmakros die alternative logische Definitionen besitzen obligatorisch.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem Symbol, das die ungültige Zuweisung vornimmt geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[078] Symbol 'symbolname' Definitionszuweisungen nicht konsistent: (mainpart/subpart 'symbolmakroname'<>'alternativdefinitionsname')!

Die für das Symbol mit dem Namen **symbolname** mit Hilfe von **$rlname**-bzw. **$rlext**-Attributeinträgen selektierte alternative logische Definition besitzt eine unterschiedliche **mainpart**/**subpart**-Zuordnung als die zum Symbolmakronamen gehörige Definition. Die **mainpart**/**subpart**-Zuordnung muss für alle zu einem Symbolmakronamen gehörigen logischen Definitionen einheitlich erfolgen (Ausnahme sind logische Definitionen mit der Spezialklasse **universal** in denen unter Umgehung von Konsistenzchecks alles erlaubt ist).

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem Symbol, das die unterschiedliche Zuweisung vornimmt geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[048] Symbol 'symbolname' hat ungueltige phys. Zuweisung!

Es wurde versucht über ein **$rpname**-, **$spname**-oder **$vgrp**-Attribut für das Symbol mit dem Namen **symbolname** einen Layoutbauteilnamen vorzugeben, der für Bauteilnamen nicht zulässige Zeichen enthält. Unzulässige Zeichen sind **?**, *****, sowie nicht druckbare Zeichencodes. Gross-/Kleinschreibung spielt keine Rolle, die Namen werden automatisch in Kleinschreibung umgewandelt. Leerzeichen werden ausgefiltert und gelten somit nicht als ungültige Zeichen. Es wird dabei nur der erste nicht aus Leerzeichen bestehende Teilstring als Namen verwendet.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[046] Symbol 'symbolname' hat ungueltigen Definitionsnamen!

Es wurde versucht, dem Symbol mit dem Namen **symbolname** über ein **$rlname**-Attribut eine alternative logische Definition zuzuweisen, deren Namen nicht zulässige Zeichen enthält. Unzulässige Zeichen sind **?** und *****, sowie nicht druckbare Zeichencodes. Gross-/Kleinschreibung spielt keine Rolle, die Namen werden automatisch in Kleinschreibung umgewandelt. Leerzeichen werden ausgefiltert und gelten somit nicht als ungültige Zeichen. Es wird dabei nur der erste nicht aus Leerzeichen bestehende Teilstring als Namen verwendet.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[047] Symbol 'symbolname' hat ungueltigen Gehaeusenamen!

Es wurde versucht dem Symbol mit dem Namen **symbolname** über ein **$plname**-Attribut eine alternatives Layoutgehäuse zuzuweisen, dessen Namen nicht zulässige Zeichen enthält. Unzulässige Zeichen sind **?** und *****, sowie nicht druckbare Zeichencodes. Gross-/Kleinschreibung spielt keine Rolle, die Namen werden automatisch in Kleinschreibung umgewandelt. Leerzeichen werden ausgefiltert und gelten somit nicht als ungültige Zeichen. Es wird dabei nur der erste nicht aus Leerzeichen bestehende Teilstring als Namen verwendet.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[111] Symbol 'symbolname' mehrfach definiert!

Das Symbol mit dem Namen **symbolname** kommt in mehreren logischen Netzlisten des Projektes vor, d.h. es ist auf mehreren Schaltplänen platziert. Vermutlich wurde ein Schaltplan mit Speichern unter in dem Projekt gespeichert oder mit Hilfe des Utilityprogrammes **COPYDDB** in das Projekt kopiert. Dabei können Namenskonflikte zwischen den Symbolen des gespeicherten/kopierten Schaltplanblattes und den Symbolen bereits in der Zieldatei vorhandener Schaltplanblätter entstehen. Zur Übernahme von Schaltplanblättern aus einer anderen Datei oder zur Kopie von Schaltplanblättern innerhalb einer Projektdatei sollte nur die Funktion Gruppe laden (in ein leeres neues Schaltplanblatt) verwendet werden, die bei Namenskonflikten eine automatische Umbenennung der Symbole entsprechend dem Symbolname Muster durchführt. Beim Gruppe laden kann auch ein normales Schaltplanblatt als Quelle selektiert werden, es ist nicht notwendig den zu kopierenden Schaltplan als Gruppe abzuspeichern. Neben Speichern unter kann auch Gruppe speichern in die Projektdatei zu solchen Namenskonflikten führen, wenn die gespeicherte Gruppe wie ein normaler Schaltplan geladen und wieder gespeichert wird. Schaltplangruppen sollten nicht in der Projektdatei, sondern in temporären anders benannten Dateien abgelegt werden.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird ein Schaltplanblatt mit einem der betreffenden Symbole geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist eines der beiden Konfliktsymbole umzubenennen. Es ist zu beachten, dass für die Bauteilattribute für die Symbole in der Projektdatei separat von den Schaltplänen unter dem Symbolnamen gespeichert werden. Dieser Eintrag enthält nach dem Zusammenkopieren mit Namenskonflikt nur die Attribute eines der beiden ursprünglichen Symbole. Beim Umbenennen sind daher auch die Attributwerte der beiden Konfliktsymbole zu überprüfen und verlorengegangene/falsche Attribute ggf. neu zu setzen.

Als Sonderfall kann diese Meldung auch bei der automatischen Generierung von Testpunkten in der Layoutnetzliste auftreten. Die Namen der Testpunktbauteile werden aus dem Testpunktnamensprefix mit angehängtem Netznamen gebildet. Für unbenannte Netze wird statt des Netznamens der Name **bauteilname.pinname** des ersten Netzpins verwendet. Der so gebildete Testpunktbauteilname wird auf 40 Zeichen beschränkt. Bei entsprechend langen Testpunktnamensprefix und ähnlichen Netz- bzw. Bauteilnamen die sich nur am Namensende unterscheiden kann es durch die Namensbeschränkung auf 40 Zeichen zu Namenskonflikten bei den automatisch generierten Testpunktbauteilnamen kommen.

[060] Symbol 'symbolname' nicht umsetzbar!

Das Symbol mit dem Namen **symbolname** konnte nicht erfolgreich in ein Layoutbauteil gepackt werden. Diese Meldung dokumentiert das Ende des gescheiterten Zuordnungsvorganges. Die genauen Gründe des Scheiterns sind in vorhergehenden Meldungen dokumentiert.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[036] Symbol 'symbolname' wurde nicht definiert!

Das Symbol mit dem Namen **symbolname** wird in der logischen Netzliste von einem Netzpin referenziert, wurde aber vorher nicht definiert. Da der Schaltplaneditor beim normalen Speichern die Symbollisten immer passend zu den Pinreferenzen generieren sollte, deutet dies auf einen Fehler auf dem Datenträger oder ein bisher unbekanntes Problem im **Schaltplaneditor** oder **Packager** hin.

Zur Problembehebung kann versucht werden die Schaltplanblätter des Projektes im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei werden die logischen Netzlisten neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[088] Symbol 'symbolname', ueberlanges $rlext 'definitionsextension' ignoriert!

Über das Attribut **$rlext** wurde eine **definitionsextension** für das Symbolmakro des Symboles **symbolname** spezifiziert, die zusammen mit dem Symbolmakronamen und dem Unterstrich die maximale Definitionsnamenslänge von 40 Zeichen überschreitet.

Dies ist nur eine Warnungsmeldung, die nicht zum Abbruch des **Packager**-Laufes führt. Für die weitere Bearbeitung wird der Wert des **$rlext**-Attributes ignoriert und der ursprüngliche logische Definitionsname verwendet.

Zur Behebung des Problemes ist ein kürzerer Wert für das Attribut **$rlext** und damit den Namen der logischen Definition zu verwenden oder zum Attribut **$rlname** zu wechseln, bei dem der Symbolmakronamen nicht zur Namensbildung herangezogen wird.

[123] Symbol : 'symbolname' Pin 'symbolpinname'

Die für den Pin **symbolpinname** des Schaltplansymbols mit dem Namen **symbolname** definierte Pinfunktion erzeugt im angeschlossenen Netz einen Konflikt bzw. Fehler, der in den vorhergehenden Meldungszeilen genauer spezifiziert ist.

[130] Symbolmakro 'symbolmakroname' nach Speichern von Blatt 'schaltplanblattname' geändert. Evtl. Pinaenderungen werden ignoriert!

Auf dem Schaltplanblatt mit dem Namen **schaltplanblattname** sind Symbole mit dem Makro **symbolmakroname** platziert, deren Makrodefinition in der Projektdatei nach dem Speichern des Schaltplanblattes geändert wurden. Die dabei evtl. vorgenommenen Änderungen an Pinnamen, Pinplatzierungen oder am Symboltagmodus werden vom **Packager** nicht berücksichtigt, da dieser nur die beim Speichern der Schaltplanblätter erzeugten logischen Netzlisten auswertet.

Zur Behebung des potentiellen Problems ist das betreffende Schaltplanblatt erneut zu speichern. Sind mehrere Schaltplanblätter betroffen kann im **Schaltplaneditor** die Funktion Einstellungen / Regelzuweisungen / Connectiviy / Projektupdate verwendet werden, die vollautomatisch alle Schaltplanblätter eines Projektes lädt und speichert und somit die vom **Packager** gelesenen logischen Netzlisten auf den neuesten Stand bringt.

[112] Symbolpin 'symbolpinname' fuer Attribut 'attributname' nicht gefunden!

Dem Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) ist in der logischen Netzliste des unter "Aktiver Planname" dokumentierten Schaltplanblattes ein Pinattribut **attributname** zugewiesen, der Symbolpin konnte aber auf dem Symbol nicht gefunden werden. Da normalerweise nur auf dem Symbolpin vorhandene Pins mit Attributen belegt sein können, deutet dies auf einen Fehler auf dem Datenträger oder ein bisher unbekanntes Problem im **Schaltplaneditor** oder **Packager** hin.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung kann versucht werden das betreffende Schaltplanblatt im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei wird die logische Netzliste neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

Wurde die logische Netzliste mit Hilfe der Anwenderfunktionen **CONCONV** oder **NETCONV** aus einem Fremdsystem importiert, ist zu überprüfen, ob die ungültige Pinattributzuweisung nicht von entsprechenden Vorgaben in der Quellnetzliste herrührt.

[081] Synth. Name von Bauteil 'symbolname' ist zu lang!

Bei der Layoutbauteilnamensbildung für ein synthetisches aus dem **architecture**-Kommando der logischen Definition des Symbols **symbolname** stammendes Bauteil wurde die maximale Namenslänge von 40 Zeichen überschritten. Zur Bildung des Layoutbauteilnamens wird das Namensmuster **[asymbolname_bauteilnummer]** verwendet. Je nach Anzahl von synthetischen Bauteilen für das Symbol und der damit verbundenen Länge der **bauteilnummer** muss man sich daher bei Symbolen mit **architecture**-Kommando auf eine Namenslänge von 33 bis 35 Zeichen beschränken.

Zur Problembehebung ist der betreffende Symbolname im Schaltplan durch Umbenennen des Symbols zu kürzen.

[007] Testpunkte Logische Bibliothek : 'definitionsname'

Diese Meldung dokumentiert den Namen der bei der automatischen Generierung von Netztestpunktbauteilen verwendeten logischen Definition. In dieser logischen Definition können Attribute und das oder die Layoutbauteilmakro(s) für die Testpunktbauteile definiert werden. Das Netz wird an den ersten Pin der logischen Definition/des Layoutbauteils angeschlossen (in der Regel gibt es bei Testpunktbauteilen ohnehin nur einen einzigen Pin).

[008] Testpunktmodus .... : modusspezifikation

Diese Meldung dokumentiert den bei der automatischen Generierung von Netztestpunktbauteilen verwendeten Bearbeitungsmodus für Netze mit nur einem Bauteilanschluss. Im Modus Alle Netze wird für jedes Netz der Netzliste ein Testpunktbauteil generiert, im Modus Keine Single-Pin-Netze hingegen werden nur Netze mit einem Testpunktbauteil versehen, die mindestens 2 Bauteilanschlüsse besitzen. Es is zu beachten, dass einzelne Netze auch explizit über das Netzattribut **$notest** von der Testpunktgenerierung ausgeschlossen werden können (zu Setzen z.B. über die Symbole **att_testdisable**, **tag_net_testdisable**, **tag_netpin_testdisable** und **tag_netarea_testdisable** aus der Bibliothek **route.ddb**).

[006] Testpunktnamensprefix .... : 'bauteilnamensprefix'

Diese Meldung dokumentiert den bei der automatischen Generierung von Testpunkten zur Bildung des Testpunktbauteilnamens dem Netznamen vorangestellten Namensprefix. Wird dieser Prefixparameter auf einen Leerstring gesetzt, so erhalten die Testpunktbauteile den Netznamen. Für unbenannte Netze wird statt des Netznamens der Name **bauteilname.pinname** des ersten Netzpins verwendet. Ein Namensprefix bietet den Vorteil, dass die Testpunktbauteile in den alphabetisch sortierten Bauteilnamensselektionsboxen des **Layouteditors** einen Block bilden und somit einfach von den anderen Bauteilen unterschieden werden können. Ein einmal eingestellter Testpunktnamensprefix wird in der Projektdatei gespeichert und muss bei folgenden **Packager**-Läufen nicht erneut angegeben werden.

[087] Treiber-Kollision auf Netz 'netzname'!

An das Netz mit dem Namen **netzname** sind entweder mehrere Pins angeschlossen, die über das Attribut **$pintype** als vom Typ **out** (d.h. Ausgänge) deklariert sind oder ein Pin vom Typ **out** und ein oder mehrere Pins der Typen **bidi** (bidirektional) oder **sup** (supply, d.h. Versorgungsspannung). In dieser Konstellation können unterschiedliche Potentiale direkt kurzgeschlossen werden. Dadurch ist der Zustand des Netzes undefiniert und die Funktionsfähigkeit der Schaltung ist zweifelhaft. Die Pins des Netzes und ihre Funktionen werden in folgenden Meldungszeilen aufgelistet.

Zur Problembehebung sind Ausgänge vom direkten Anschluss an das Netz zu trennen oder anders zu deklarieren, falls sie nicht ständig treibend sind.

[058] Ungueltige Swapdaten fuer Pin 'layoutbauteilpinname'!

In der logischen Definition des in der vorhergehenden Meldungszeile aufgelisteten Symbols sind widersprüchliche **swap**-Kommandos für den Layoutbauteilpin mit dem Namen **layoutbauteilpinname** enthalten. Da beim Einspielen der logischen Definition für ein Symbol ein Konsistenzcheck für die **swap**-Kommandos durchgeführt wird, stammen die widersprüchlichen **swap**-Kommandos evtl. aus den unterschiedlichen Definitionen einer **mainpart**/**subpart**-Kombination oder aus einer sehr alten BAE-Version ohne Konsistenzcheck.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung sind die **swap**-Kommandos für den Layoutbauteilpin in der/den logischen Definition(en) zu korrigieren.

[001] Ungueltiger Aufruf!

Diese Fehlermeldung erscheint an der Konsole, bzw. in einem Terminalfenster, wenn in der **DOS** bzw. in einer der **UNIX/Linux**-Versionen versucht wird, den **Packager** (**logpack.exe** bzw. **logpack**) direkt und nicht aus der BAE-Benutzeroberfläche heraus aufzurufen. Ein direkter Aufruf ist nicht erlaubt, da der Wechsel in andere BAE-Programmmodule nicht funktionieren würde und dem **Packager** die Pfade zu den Setup-Dateien/Bibliotheken nicht bekannt wären.

[051] Ungueltiges logisches Bauteilformat : 'symbolname'!

Die von dem Symbol mit dem Namen **symbolname** referenzierte logische Definition enthält Optionen, die vom **Packager** nicht verstanden werden. Dies kann von einem Fehler auf dem Datenträger oder der Verwendung einer neueren BAE-(Zwischen)Version beim Einspielen der logischen Definitionen herrühren.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung kann versucht werden die logische Definition erneut in das Projekt bzw. die verwendete Bibliothek einzuspielen bzw. neu zu editieren. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[024] Ungueltiges logisches Netzlistenformat!

Die logischen Netzlisten der Projektdatei enthalten Optionen, die vom **Packager** nicht verstanden werden. Dies kann von einem Fehler auf dem Datenträger oder der Verwendung einer neueren BAE-(Zwischen)Version beim Speichern der Schaltpläne herrühren.

Zur Problembehebung kann versucht werden die Schaltplanblätter des Projektes im **Schaltplaneditor** zu laden und erneut zu speichern. Dabei werden die logischen Netzlisten neu generiert. Sollte dies nicht zur Behebung des Problems führen empfiehlt es sich die Projektdatei zur näheren Untersuchung an den Bartels-Support zu senden.

[119] Variante variantennummer Untersymbol 'symbolname' platziert!

Das angegebene Symbol mit dem Namen **symbolname** ist mit anderen Untersymbolen zusammen in das in der vorhergehenden Meldungszeile gelistete Layoutbauteil gepackt worden und hat das **$noplc**-Attribut auf einen Leerstring gesetzt, während an den anderen Untersymbolen das **$noplc**-Attribut gesetzt ist. Der **Packager** gibt hierbei der Platzierungsanforderung Vorrang, wodurch die anderen als unplatziert markierten Untersymbole mitplatziert werden. Es bleibt dem Benutzer überlassen zu entscheiden, ob dies die Funktion der Schaltung beeinträchtigt.

Zur Problembehebung ist das **$noplc**-Attribut in den Untersymbolen ggf. auf einheitliche Werte zu setzen.

[118] Variante variantennummer Untersymbol 'symbolname' unplatziert!

Das angegebene Symbol mit dem Namen **symbolname** ist mit anderen Untersymbolen zusammen in das in der vorhergehenden Meldungszeile gelistete Layoutbauteil gepackt worden und hat das **$noplc**-Attribut gesetzt, während an den anderen Untersymbolen das **$noplc**-Attribut auf einen Leerstring gesetzt ist. Der **Packager** gibt hierbei der Platzierungsanforderung Vorrang, wodurch das gelistete Untersymbol mitplatziert wird. Es bleibt dem Benutzer überlassen zu entscheiden, ob dies die Funktion der Schaltung beeinträchtigt.

Zur Problembehebung ist das **$noplc**-Attribut in den Untersymbolen ggf. auf einheitliche Werte zu setzen.

[062] Variantenbauteil 'layoutbauteilname'/'symbolname': Konflikt mit $rpname (Liste:)

Das Symbol mit dem Namen **symbolname** wurde über den Wert des **$vgrp**-Attributes in das variantenabhängig belegtes Layoutbauteil mit dem Namen **layoutbauteilname** eingebettet. Der ursprüngliche **symbolname** wird nun seinerseits von den in der folgenden Liste aufgeführten Schaltplansymbolen über das **$rpname**-Attribut als Layoutbauteilname angefordert (für ein vom Variantenbauteil verschiedenes Layoutbauteil). Dies kann zu Inkonsistenzen in der Attributverwaltung führen und ist daher nicht erlaubt.

Zur Problembehebung ist das Symbol mit dem Namen **symbolname** umzubenennen oder für die gelisteten Konfliktsymbole im **$rpname**-Attribut ein anderer Layoutbauteilname vorzugeben. Es ist zu empfehlen Symbole die über das Attribut **$vgrp** zur die Verwendung in variantenabhängigen Layoutbauteilen vorgesehen sind mit einem "exotischen" Symbolname Muster zu versehen, das nicht mit gängigen Layoutbauteilnamen kollidiert. Für diese Symbole spielt der Symbolname ohnehin eine untergeordnete Rolle, da der spätere Layoutbauteilname durch das **$vgrp**-Attribut festgelegt wird.

[061] Variantensymbol 'symbolname': xlat mit Gattern nicht erlaubt!

Das Symbol mit dem Namen **symbolname** wurde über ein **$vgrp**-Attribut in ein variantenabhängig belegtes Layoutbauteil eingebettet und durch ein auf einen Leerstring gesetztes **$noplc**-Attribut als variantenabhängig bestückt markiert. Dies ist für Mehrfachgatter (d.h. Symbole, die in der logischen Definition über das **xlat**-Kommando in mehreren Instanzen in ein Layoutbauteil gepackt werden) nicht erlaubt.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

[034] Virtuelle Variantenbauteile nicht erlaubt ('symbolname')!

Das in seiner logischen Definition als **virtual** deklarierte Symbol mit dem Namen **symbolname** wurde über ein **$vgrp**-Attribut in ein variantenabhängig belegtes Layoutbauteil eingebettet und durch ein auf einen Leerstring gesetztes **$noplc**-Attribut als bestückt markiert. Da ein **virtual**-Symbol nicht in die Layoutnetzliste übertragen wird, ist die Verwendung von **$vgrp**-/ **$noplc**-Attributen hier nicht sinnvoll.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Behebung des Problems sind die **virtual**-Deklaration aus der logischen Netzliste zu entfernen oder die Werte der **$vgrp**/**$noplc**-Attribute zurückzusetzen und ggf. DANACH die **$vgrp**-/ **$noplc**-Texte vom Symbol zu entfernen um weitere Fehlzuweisungen zu verhindern.

[127] Widerspruechliche $netname-Netznamensvorgaben 'netzname1'<>'netzname2'!

Einem Netz wurden über Netznamensattribute **$netname** die beiden unterschiedlichen Netznamen **netzname1** und **netzname2** zugewiesen.

Zur Problembehebung ist eine der Netznamenszuweisungen zurückzunehmen.

[138] Ziellayout 'layoutname' ist vom Benutzer 'benutzername' geladen.

Das zum selektierten Netzlistennamen gehörige Layout **layoutname** ist gerade vom Benutzer **benutzername** in den **Layouteditor** geladen. Da hierbei die (alte) Netzliste mitgeladen ist, gefährdet diese Situation die Datenkonsistenz nach dem **Packager**-Lauf, da beim Speichern des Layouts die alte Netzliste mitgespeichert wird und damit ggf. die durch den aktuellen **Packager**-Lauf erzeugte Netzliste wieder überschreibt. Zur Vermeidung dieser Situation ist den weiteren Anweisungen der Warnungsmeldung zu folgen und zunächst das Layout zu speichern um dieses dann nach dem **Packager**-Lauf mit der neu erzeugten Netzliste wieder zu laden.

Die Information zum aktuellen Bearbeiter des Layouts wird aus der zum Projekt gehörigen **.lck**-Datei abgeleitet. Sollte diese von unkontrolliert beendeten früheren Sitzungen stehen geblieben sein und zu unnötigen Warnungen über nicht mehr aktive Bearbeiter führen, kann die **.lck**-Datei problemlos gelöscht werden, sie enthält keine Designdaten.

[016] Zu wenig Speicherplatz!

Es konnte nicht genügend Hauptspeicher für die Bearbeitung der Projektdaten allokiert werden. Diese Fehlermeldung sollte bei der Speicherausstattung heutiger Rechner nicht mehr erscheinen. Zumindest nicht, wenn der Rechner nicht durch parallel ablaufende speicherfressende Programme ausgelastet ist.

[121] Zuordnung fuer $gp 'layoutbauteilpinname' nicht gefunden!

Dem in der Folgezeile dokumentierten Symbol wurde über das **$gp**-Attribut ein Layoutbauteilpinname zugeordnet, der in der logischen Definition in keinem der über das **xlat**-Kommando definierten Gatter als erster Pin aufgeführt ist.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist das **$gp**-Attribut auf einen in der logischen Definition als ersten Gatterpin aufgeführten Layoutbauteilpinnamen zu setzen oder die logische Definition zu korrigieren.

[063] Zuweisungs-Pin 'symbolpinname' nicht gefunden!

Der Symbolpin mit dem Namen **symbolpinname** (der Symbolname ist in der vorhergehenden Meldungszeile dokumentiert) wurde in der logischen Definition über ein **netattr**-Kommando als Zielpin für ein Netzattributname angegeben, wurde auf dem Symbol aber nicht gefunden.

Bei unmittelbarem Wechsel in den **Schaltplaneditor** wird das Schaltplanblatt mit dem betreffenden Symbol geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der Pin auf dem Symbol zu platzieren bzw. in der logischen Definition zu korrigieren. Es ist zu beachten, dass nach dem Platzieren/Löschen/Umbenennen von Symbolpins die Schaltpläne mit Platzierungen des bearbeiteten Symbols geladen und gespeichert werden müssen um die vom **Packager** bearbeiteten logischen Netzlisten auf den neuesten Stand zu bringen. Die Symbolmakros selbst werden vom **Packager** nicht gelesen. Beim einfachen Laden der Schaltpläne sieht man die neuen/geänderten Pinnamen zwar auf dem Bildschirm, in die logische Netzliste werden die Daten aber erst beim Speichern der Schaltplanblätter übertragen.

[059] xlat-Pin 'layoutbauteilpinname' auf dem Layoutbauteilmakro nicht vorhanden!

Die von dem in der vorhergehenden Meldungszeile aufgelisteten Symbol angeforderte logische Definition enthält in einem **xlat**-Kommando eine Zuordnung zu einem Layoutbauteilpin mit dem Namen **layoutbauteilpinname** der auf dem verwendeten Layoutbauteilmakro nicht gefunden werden konnte.

Abhängig vom eingestellten Modus für die Fehlerbehandlung ist dies eine Warnungs- oder eine Fehlermeldung. Im Falle einer Warnung setzt der **Packager** die Bearbeitung fort, als ob es den Pin auf dem Layoutbauteilmakro gäbe und man kann diesen nachträglich auf dem Layoutbauteilmakro platzieren um ein zur Netzliste passendes Layout zu erhalten. Im Falle eines Fehlers wird bei einem unmittelbarem Wechsel in den **Schaltplaneditor** das Schaltplanblatt mit dem Symbol, das das Layoutbauteil referenziert, geladen und ein Zoom Fenster an die Symbolposition durchgeführt.

Zur Problembehebung ist der fehlende Pin auf dem Layoutbauteilmakro zu platzieren oder im **xlat**-Kommando der logischen Definition durch einen auf dem Layoutbauteilmakro vorhandenen Pin zu ersetzen.

3.3 Backannotation

Backannotation ist immer dann durchzuführen, wenn im Layout Änderungen an der Netzliste (Umbenennung von Bauteilen, Pin/Gate Swaps oder Zuweisung alternativer Gehäusebauformen) vorgenommen wurden, die in den Stromlauf zurückgemeldet werden müssen. Bauteile können mit der Funktion Name in Netzliste im Menü zur manuellen Bauteilplatzierung umbenannt werden. Pin/Gattertausch kann entweder interaktiv mit der Funktion Pin/Gate Swap im Menü zur manuellen Bauteilplatzierung oder automatisch mit den Funktionen Voll-Autoplace, Einzeloptimierung bzw. Multi-Optimierung im **Autoplacement** durchgeführt werden. Darüber hinaus kann auch die Option Router P/G-Swap Ein im **Autorouter** die automatische Anwendung von Pin/Gate Swaps während des Rip-Up-Routings bewirken. Die Zuweisung alternativer Gehäusebauformen erfolgt mit der Untermenüfunktion Bauform aendern während der manuellen Bauteilplatzierung im Layout. Es ist dringend zu beachten, dass alle oben genannten Modifikationen in der physikalischen Netzliste verloren gehen, wenn sie vor einer neuerlichen Schaltplanbearbeitung nicht mit Backannotation in den Stromlauf zurückgemeldet werden.

3.3.1 Funktionsaufruf

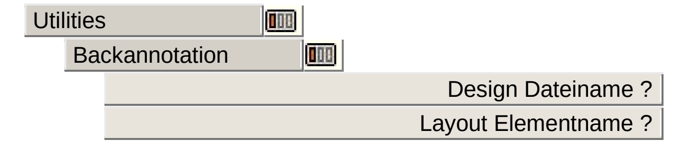
Die Backannotation ist vollständig im **Schaltplaneditor** integriert und kann über das Kommando Backannotation aus dem Menü Diverse bzw. Utilities gestartet werden.

Automatischer Aufruf der Backannotation

Die SCM-Funktionen zum Laden von Elementen sind mit einem Mechanismus zur wahlweisen automatischen Durchführung notwendiger Backannotation-Prozesse beim Laden von Stromlaufplänen in den **Schaltplaneditor** ausgestattet. Hierzu werden layoutspezifische Datenbankeinträge ausgewertet, die im Layoutsystem beim Speichern generiert werden, wenn Pin-/Gate-Swaps oder Bauteilnamensänderungen vorgenommen wurden. Ist beim Laden eines SCM-Plans eine Anforderung zur Durchführung der Backannotation vorhanden, dann wird automatisch eine Bestätigungsabfrage aktiviert, die wahlweise die Durchführung der Backannotation ermöglicht. Nach dem Backannotation-Lauf wird der Datenbankeintrag für die Backannotation-Anforderung wieder gelöscht.

3.3.2 Funktionsablauf

Nach dem Aufruf der Backannotation wird der Benutzer nacheinander um den Namen der Design- bzw. Projektdatei und den Namen des Layouts gefragt, wobei durch Betätigung der ESC-Taste (ASCII-Code 003) ein Abbruch des Backannotation-Laufs veranlasst werden kann:



Für den Design Dateinamen ist der Name der Projektdatei einzugeben, für die die Netzlistenumsetzung durchgeführt werden soll. Die Projektdatei muss mit der Extension `.ddb` verfügbar sein; der Dateiname ist ohne diese Extension anzugeben. Wird ein Leerstring (Betätigen der Eingabetaste ↵) für den Dateinamen eingegeben, dann verwendet das System den Dateinamen des aktuell geladenen bzw. des im vorhergehenden Programm-Modul des **Bartels AutoEngineer** geladenen Elements, d.h. den systemweiten Projektnamen.

Auf die Abfrage nach dem Layout Elementnamen ist der Name der zu transferierenden Netzliste bzw. des modifizierten Layouts einzutragen. Betätigt der Anwender hier nur die Eingabetaste ↵(leerer String), dann trägt die Software automatisch den über das Setup (siehe Kommando **`LAYDEFELEMENT`** in **BSETUP**-Beschreibung) definierten Default-Layoutelementnamen ein.

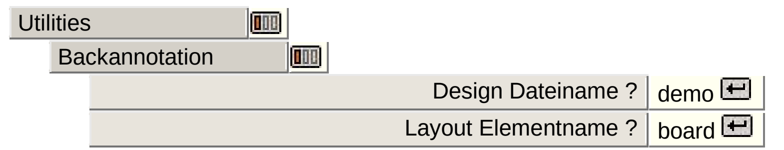
Nach der Eingabe des Layout Elementnamen wird die Layoutnetzliste in den Stromlaufplan zurücktransferiert. Unter dem Projektdateinamen mit der Extension `.ass` wird eine Assignmentlist ausgegeben. Die Assignmentlist wird vom System nicht weiter benötigt und ist zur Auswertung durch den Benutzer bestimmt (Änderungshistorie).

Nach erfolgreichem Backannotation-Lauf erscheint die Meldung **`Es wurden keine Fehler festgestellt.`**, und die Backannotation-Funktion kann durch durch Betätigung einer beliebigen Taste beendet werden. Sollte es zu Fehlermeldungen kommen, dann deutet dies in der Regel auf die falsche Angabe des Layoutelementnamens oder das Fehlen der logischen Netzliste hin. Eine erfolgreiche Druchführung der Backannotation ist auch nicht möglich, wenn erforderliche Schaltplansymbole bzw. Bauteile fehlen weil sie zuvor irrtümlicherweise aus dem Schaltplan gelöscht wurden. Die Backannotation gibt ggf. Hinweise auf fehlende Bauteile aus. Diese Bauteile sind im Schaltplan zu platzieren, um anschließend einen erfolgreichen Backannotation-Lauf zu ermöglichen (das Problem fehlender SCM-Bauteile liesse sich auch durch einen **Packager**-Lauf "beheben", wodurch aber dann ggf. Layoutmodifikationen wie Pin-/Gateswaps verloren gehen würden).

3.3.3 Beispiel

Im Folgenden soll ein Backannotation-Lauf für die Projektdatei **demo.ddb** durchgeführt werden. Wechseln Sie hierzu in das Verzeichnis, in dem **demo.ddb** abgelegt ist.

Starten Sie BAE, rufen Sie den **Schaltplaneditor** auf, und transferieren Sie mit den folgenden Kommandos die in **demo.ddb** enthaltene physikalische Netzliste mit dem Namen **board** zurück in den Stromlauf:



Backannotation gibt folgende Meldungen auf dem Bildschirm aus:

```
==============================
BARTELS BACKANNOTATION UTILITY
==============================

Design Dateiname ........: 'demo'
Layout Elementname ......: 'board'
Es wurden keine Fehler festgestellt.
```

Der Backannotation-Lauf wurde erfolgreich beendet (Meldung **Es wurden keine Fehler festgestellt.**), und die in der Datei **demo.ddb** enthaltene logische Netzliste wurde ggf. modifiziert.

Die von der Backannotation erzeugte Assignmentslist **demo.ass** wird vom System nicht weiter benötigt. Sie können mit Hilfe Ihres Editors einen Blick auf die in dieser Datei enthaltenen Informationen werfen, um festzustellen, welche Bauteilnamen geändert wurden bzw. welche Pins oder Gatter gegeneinander getauscht wurden.

Backannotation bildet die modifizierte Netzliste auf den Stromlaufplan ab. Überprüfen Sie dies, indem Sie sich nach dem erfolgreichen Backannotation-Lauf im Stromlauf-Editor die entsprechenden Stromlaufblätter ansehen.

3.4 Utilities zur Netzlistenverarbeitung

Die Netzliste wird im **Bartels AutoEngineer** üblicherweise mit dem Stromlauf-Editor erzeugt. Es besteht daneben aber auch die Möglichkeit, (in Fremdsystemen erzeugte) ASCII-Netzlisten in den **AutoEngineer** einzuspielen. Der **Bartels AutoEngineer** enthält die nachfolgend aufgeführten Utilityprogramme (genaue Beschreibung dieser Utilities siehe Kapitel 7), mit deren Hilfe Netzlisten verschiedener Formate eingelesen bzw. ausgegeben werden können.

3.4.1 Einlesen logischer Netzlisten

Sofern logische, ungepackte Netzlisten in den **AutoEngineer** eingespielt werden, ist anschließend ein **Packager**-Lauf durchzuführen, bevor das Layout erstellt werden kann. Dies setzt natürlich das Vorhandensein einer Bibliothek mit den vom **Packager** benötigten Informationen voraus. D.h., ggf. ist vor dem **Packager**-Lauf eine DDB-Datei mit allen benötigten Layoutsymbolen und **LOGLIB**-Einträgen zu erstellen.

Das Programm **NETCONV** (Logical Netlist Conversion Utility) dient dazu, logische (d.h. ungepackte) Netzlisten in den **Bartels AutoEngineer** zu übertragen.

3.4.2 Einlesen physikalischer Netzlisten

Bei der Umsetzung bereits gepackter Netzlisten in den **AutoEngineer** erübrigt sich der **Packager**-Lauf. Allerdings wird das Vorhandensein einer Bibliothek mit den vom Layout benötigten Informationen vorausgesetzt. D.h., ggf. ist vor der Netzlistenumsetzung eine DDB-Datei mit allen benötigten Layoutsymbolen zu erstellen.

Das Programm **CONCONV** (Connections Conversion Utility) erlaubt die Übertragung von ASCII-Netzlisten im Bartels-, CALAY-, MARCONI- oder RACAL-Format in den **AutoEngineer**.

Das Programm **REDASC** (REDAC ASCII Input Interface) erlaubt die Übernahme von Layoutdaten des Redac-MAXI-Systems, d.h. sowohl die Transformation von Layoutsymbolen, als auch die Übertragung der Bauteilliste, der Netzliste, sowie ggf. der Platzierung im MAXI-CDI-Format in den **AutoEngineer**.

3.4.3 Netzlistenausgabe

Die Ausgabe von Netzlisten beliebiger Formate ist z.B. mit Hilfe des Programms **USERLIST** möglich. Hierzu ist die Definition eines **USERLIST**-Scripts notwendig. Dieses Script wird vom Programm **USERLIST** interpretiert, und es wird entsprechend den Vorgaben im Script eine ASCII-Ausgabedatei generiert. Erstellen Sie mit Ihrem Editor das **USERLIST**-Script `netlist.usf` mit folgendem Inhalt:

```
EXTENSION = ".nl";
PRINT("NETLIST ",PROJECTNAME,";",CR);
FOR (ALL PARTS)
{
    PRINT(PARTNAME:-5,"(",$plname:-8,") : ");
    CLEARCOUNTER;
    FOR (ALL PINS)
    {
        IF (NETPINCOUNT > 1)
        {
            PRINT(PINNAME,"(",NETNAME,")");
            COUNTUP;
            IF (COUNTVALUE > 4) { PRINT(CR,TAB);CLEARCOUNTER; }
        }
    }
    PRINT(";",CR);
}
ENDSPEC
```

Nun können Sie mit folgendem **USERLIST**-Aufruf die in `demo.ddb` gespeicherte physikalische Netzliste `board` auf die Datei `demo.nl` ausgeben:

> **userlist netlist demo board** ↵

Nach erfolgreichem **USERLIST**-Lauf sollte die Datei `demo.nl` folgenden Inhalt haben:

```
NETLIST board;
c100 (chip1210) : 1(vss)2(vdd);
c101 (chip1206) : 1(vss)2(net);
ic10 (dil14   ) : 1(@7)10(@9)11(@2)12(@3)13(@9)14(vdd)2(net)
    3(@6)4(@4)5(@5)6(@6)7(vss)8(@11)9(@4);
k1   (relais  ) : a1(vdd)a2(@14)c1(vdd)c2(vdd)nc1(bus.out1)
    nc2(bus.out3)no1(bus.out0)no2(bus.out2);
r100 (r04a25  ) : 1(@7)2(@6);
r101 (r04a25  ) : 1(@5)2(@4);
r102 (r04a25  ) : 1(@11)2(@9);
r103 (r04a25  ) : 1(@3)2(@2);
r104 (minimelf) : 1(@2)2(@8);
r105 (chip1206) : 1(net)2(vdd);
s1000(s1dilo  ) : 1(@7)2(net);
s1001(s1dilo  ) : 1(@5)2(net);
s1002(s1dilo  ) : 1(@11)2(net);
s1003(s1dilo  ) : 1(@3)2(net);
s1004(s1dilo  ) : 1(net)2(vss);
s1005(s1dilo  ) : 1(net)2(vss);
s1006(s1dilo  ) : 1(net)2(vss);
s1007(s1dilo  ) : 1(net)2(vss);
s1008(s1dilo  ) : 1(net)2(vss);
s1009(s1dilo  ) : 1(net)2(vss);
v1   (to92    ) : 1(vss)2(@8)3(@14);
v1000(d04a25  ) : a(@14)k(vdd);
x1000(xsubd9bl) : 1(vss)4(bus.out0)5(bus.out1)6(bus.out2)7(bus.out3)
    9(vdd);
```

Mit folgendem **USERLIST**-Aufruf lässt sich die in `demo.ddb` gespeicherte logische Netzliste `board_log` auf die Datei `demo.nl` ausgeben:

```
> userlist netlist demo board_log ↵
```

Nach erfolgreichem **USERLIST**-Lauf sollte die Datei **demo.nl** folgenden Inhalt haben (beachten Sie die im Vergleich zur vorherigen Ausgabedatei unterschiedlichen Pinbezeichnungen und **$plname**-Einträge):

```
NETLIST board_log;
c100 (c       ) : 1(vss)2(vdd);
c101 (c       ) : 1(vss)2(net);
ic10 (cd4081  ) : a(@7)b(net)y(@6);
ic11 (cd4081  ) : a(@5)b(@6)y(@4);
ic12 (cd4081  ) : a(@11)b(@4)y(@9);
ic13 (cd4081  ) : a(@3)b(@9)y(@2);
k10  (rels    ) : a1(vdd)a2(@14);
kk100(relc    ) : c(vdd)nc(bus.out1)no(bus.out0);
kk101(relc    ) : c(vdd)nc(bus.out3)no(bus.out2);
r100 (r       ) : 1(@7)2(@6);
r101 (r       ) : 1(@5)2(@4);
r102 (r       ) : 1(@11)2(@9);
r103 (r       ) : 1(@3)2(@2);
r104 (r       ) : 1(@2)2(@8);
r105 (r       ) : 1(net)2(vdd);
s1000(s_1dil  ) : 1(@7)2(net);
s1001(s_1dil  ) : 1(@5)2(net);
s1002(s_1dil  ) : 1(@11)2(net);
s1003(s_1dil  ) : 1(@3)2(net);
s1004(s_1dil  ) : 1(net)2(vss);
s1005(s_1dil  ) : 1(net)2(vss);
s1006(s_1dil  ) : 1(net)2(vss);
s1007(s_1dil  ) : 1(net)2(vss);
s1008(s_1dil  ) : 1(net)2(vss);
s1009(s_1dil  ) : 1(net)2(vss);
v1   (tr_bc517) : b(@8)c(@14)e(vss);
v1000(d       ) : a(@14)k(vdd);
x1000(x_subd9b) : 1(vss)4(bus.out0)5(bus.out1)6(bus.out2)7(bus.out3)
    9(vdd);
```

3.4.4 Netzattribute

Im **Packager** sind Funktionen zur Übernahme beliebiger netzspezifischer Attribute integriert. Diese Netzattribute können im Stromlaufplan an speziellen Netzattributsymbolen definiert werden (Funktion Wert zuweisen im Menü Symbole). Damit der **Packager** diese Information in die physikalische Netzliste übertragen kann, ist zusätzlich ein entsprechender Eintrag in der Logischen Bibliothek erforderlich. Abbildung 3-3 zeigt ein Beispiel für die Definition und Verwendung von Netzattributsymbolen mit den entsprechenden Loglib-Definitionen (siehe hierzu auch die Beschreibung des Utilityprogramms **LOGLIB** im Kapitel 7.11 dieses Handbuchs).

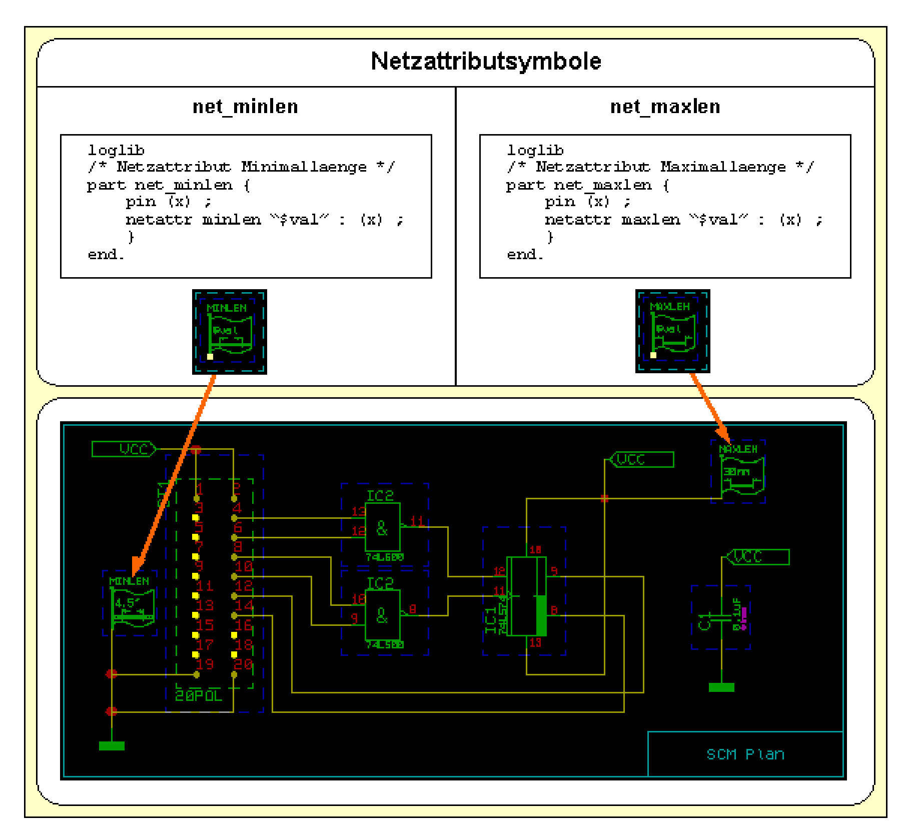


Abbildung 3-3: Netzattribut-Definitionen

Kapitel 4
Leiterkartenentwurf / CAD

Dieses Kapitel erläutert die Handhabung der Programm-Module **Layouteditor** und **Autorouter** für den Entwurf des Leiterkartenlayouts sowie der Programm-Module **CAM-Prozessor** und **CAM-View** zum Erstellen und Bearbeiten der Fertigungsdaten für die Leiterkartenproduktion. Hierbei wird der Leser anhand von Beispielen in einer logischen Abfolge durch die Erstellung von Bibliothekselementen sowie die Bearbeitung und das Layout von Leiterplattendesigns bis hin zu den prozeduren für das Erstellen und Bearbeiten der Fertigungsdaten für die Leiterkartenherstellung geführt. Das in diesem Kapitel auf der Basis der Designdaten aus den vorhergehenden Kapiteln erstellte Leiterkartenlayout wird in den nachfolgenden Kapiteln einer weiteren Bearbeitung unterzogen. Es wird daher empfohlen, dieses Kapitel Schritt für Schritt und ohne Auslassung irgendwelcher Abschnitte durchzuarbeiten, um einen vollstädigen Überblick über den Funktionsumfang des Layoutsystems zu gewinnen. Sobald ein spezielles Kommando angewandt bzw. dessen Benutzung erläutert wurde, ist davon auszugehen, dass der Leser dieses Kommando verstanden hat und es bei Bedarf ohne nähere Erläterungen wieder ausführen kann. Nachfolgende Instruktionen zu dem betreffenden Kommando sind dann weniger ausführlich, um das Lesen zu vereinfachen und den Lernprozess zu beschleunigen.

Inhalt

Tabellen

Abbildungen

4.1 Allgemeine Hinweise

Das Leiterkartenlayoutsystem des **Bartels AutoEngineer** besteht im Wesentlichen aus einem grafisch-interaktiven **Layouteditor** mit integriertem Layoutsymboleditor und integrierten Funktionen zur automatischen Bauteilplatzierung und zur automatischen Generierung von Füllflächen, dem **Bartels AutoEngineer** zur vollautomatischen Entflechtung, sowie dem **CAM-Prozessor** und dem **CAM-View**-Modul zur Erstellung und Bearbeitung der Fertigungsdaten für die Leiterkartenproduktion. Die nachfolgenden Abschnitte dieses Benutzerhandbuchs enthalten eine detaillierte Beschreibung dieser Programm-Module und deren Anwendung.

4.1.1 Komponenten und Leistungsmerkmale

Layouteditor (Grafikeditor)

Die Fließpunktarithmetik des grafischen **Layouteditors** ermöglicht, Daten metrisch oder in Zollwerten frei zu definieren, dadurch jede erdenkliche Form für Lötaugen und Kupferflächen zu generieren und diese frei zu positionieren (z.B. Drehungen um einen bestimmten, auf acht Stellen hinter dem Komma vorgegebenen Winkel). Lötaugen, Bauteile und geometrische Gebilde werden also nicht nur annäherungsweise platziert, sondern sitzen exakt da, wo sie hingehören. Kreise sind auch auf dem Bildschirm echte Kreise und keine elliptischen Gebilde. Die Sichtbarkeit von Flächen lässt sich wahlweise vom Spiegelungsmodus abhängig machen, d.h. es können z.B. SMD-Anschlüsse definiert werden, die auf der Bauteilseite eine andere Form haben als auf der Lötseite.

Der Online-Check ermittelt rasterunabhängig auf acht Stellen hinter dem Komma die genauen Abstandswerte. Mit der gleichen Genauigkeit werden Abstandsverletzungen registriert und umgehend auf dem Bildschirm angezeigt. So können Fehlerquellen sofort erkannt und eliminiert werden, nachträgliche, komplizierte Korrekturen entfallen. Der Online-Check arbeitet *inkremental* und daher mit extrem hoher Geschwindigkeit (d.h. in Echtzeit!).

Die zwanzigstufige Undo/Redo-Funktion ermöglicht auch hier, Alternativen parallel durchzuspielen. Beliebige Teile der Leiterplatte können - zu Gruppen zusammengefasst - bewegt, gedreht, gespiegelt, kopiert und archiviert werden. Gleiches gilt für Leiterbahnstrukturen ohne Bauteile. Das spart, beispielsweise beim Erstellen spezieller Busstrukturen, eine Menge Zeit.

Das schnelle interaktive Platzieren von Bauteilen unter ständiger Aktualisierung der Verbindungen garantiert eine optimale Ausnutzung der Leiterplattenfläche. Während das Bauteil auf dem Bildschirm bewegt wird, aktualisieren sich die Verbindungen ohne Zeitverlust dynamisch zum nächst gelegenen Anschlusspunkt. Alle Bauteile lassen sich für die SMD-Anwendung spiegeln und um jeden beliebigen Winkel drehen. Es können beliebige Koordinaten für die Bauteilplatzierung spezifiziert werden und auch Polarkoordinaten für die Platzierung von Bauteilen auf einem Kreisbogen sind möglich. Ein komfortabler Automatismus ergibt sich aus der Möglichkeit, Defaultwerte für die Drehung (in 90-Grad-Schritten) und Spiegelung der Bauteile vorzugeben. Während des manuellen Platzierens kann eine alternative Gehäusebauform für das aktuell bearbeitete Bauteil gewählt werden.

Da das System netzorientiert und nicht pinorientiert ist, können Verbindungen nicht nur über Leiterbahnen, sondern auch über Kupferflächen hergestellt werden. Es muss also nicht extra ein Anschlusspin oder eine Durchkontaktierung angefahren werden (echte Connectivity).

Leiterbahnen und Kupferflächen können mit Fließpunktgenauigkeit in jedem beliebigen Raster erstellt werden. Das gewünschte oder interaktiv bearbeitete Potential wird sofort "gehighlighted", das heißt aufgehellt dargestellt, und ist an jedem beliebigen Punkt des gleichen Netzes anschließbar. Die Verwendung von partiellen Durchkontaktierungen (Blind und Buried Vias) ist beim Verlegen von Leiterbahnen ebenso möglich, wie die Definition von kreisbogenförmigen Leiterbahnsegmenten.

Das System bietet die Möglichkeit der Definition von Versorgungslagen. Auf diesen Versorgungslagen können Potentialflächen in beliebiger Form platziert werden. Damit lassen sich geteilte Potential- bzw. Versorgungslagen (Split Power Planes) erzeugen.

Ausgesprochen nützlich für analoge bzw. gemischt analog-digitale Designs sind die Funktionen zum automatischen Füllen von Kupferflächen. Dabei kann mit einstellbarem Isolationsabstand, definierbarer minimaler Strukturgröße, sowie der wahlweisen Elimination isolierter Potentialflächen gearbeitet werden. Wärmefallen werden nach Bedarf automatisch erzeugt, wobei auch die Anschlussbreite einstellbar ist. Elektrisch leitfähige Flächen lassen sich in linien- oder gitterschraffierte (Schirm-)Flächen mit definierbarer Schraffurbreite und vorgebbarem Schraffurabstand umwandeln.

Besonders hilfreich für HF- und Analog-Anwendungen ist die Möglichkeit, Kupferflächen stufenlos zu vergrößern bzw. zu verkleinern. Segmentlängen von Leiterbahnen und Kantenlängen von Kupferflächen können jederzeit automatisch abgefragt werden. Konstruktionselemente werden zu Dokumentationszwecken vom System bemaßt.

Durch die Einbindung der **Bartels User Language** in den **Layouteditor** hat der Anwender die Möglichkeit, eigene Menüfunktionen (Makros), Postprozessoren, Report-, Test- und Editierfunktionen, usw. zu implementieren, die er wahlweise explizit (über eine spezielle Menüfunktion) oder implizit (über Tastatur oder ereignisgesteuert) aktivieren kann.

Autoplacement

Das BAE-Layoutsystem ist mit mächtigen **Autoplacement**-Verfahren ausgestattet. Vor der automatischen Platzierung können die zu platzierenden Bauteile nach dem Mengenprinzip selektiert werden. Die Auswahl der Bauteile erfolgt durch die Spezifikation der Bauteil- und Bibliotheksteilnamen, wobei auch Namensmuster (Wildcards) zulässig sind. Die selektive Auswahl von Bauteilen aus einem speziellen Blockschaltbild eines hierarchischen Schaltplanentwurfs ist ebenfalls möglich. Die Auswahl der Bauteilmenge bietet dem Anwender die Möglichkeit der Steuerung des Platzierungsablaufs (z.B. erst die Stecker, dann DIL-Gehäuse, dann Abblockkondensatoren, usw.).

Mit dem Matrixplacement-Verfahren kann eine selektierbare Menge von Bauteilen automatisch auf einem definierbaren Einbauplatzraster platziert werden. Dadurch reduziert sich der Aufwand für die Platzierung gleichartiger Bauteiltypen (Speicherbausteine, Abblockkondensatoren, Testpunkte, usw.) ganz erheblich. Selbstverständlich berücksichtigt die Matrix-Platzierungsfunktion auch die Defaulteinstellungen für die Drehung und Spiegelung der Bauteile.

Die integrierten Initialplacement-Funktionen ermöglichen die vollautomatische Durchführung der Bauteilplatzierung. Hierbei werden die unplatzierten Bauteile innerhalb der Platinenumrandung auf dem aktuell eingestellten Platzierungsraster platziert, wobei vorplatzierte Bauteile (Stecker, LEDs, etc.) ebenso berücksichtigt werden wie die Vorgaben aus der Netzliste. Abblockkondensatoren und SMD-Bauteile werden automatisch erkannt. Die Lötseite kann für die SMD-Platzierung wahlweise gesperrt oder freigegeben werden. Die Bauteile werden selbsttätig in 90-Grad-Schritten gedreht, wobei sich die Freiheitsgrade bei der Bauteilrotation für eine fehlersichere Bestückung wahlweise einschränken lassen. Optional kann ein Bauteilexpansionsparameter zur Generierung von Freiflächen zwischen den Bauteilen definiert werden. Die automatische Platzierung wird durch einstellbare Gewichtungsfaktoren zur Berücksichtigung von Netzlistenvorgaben und zur Bewertung der Bauteil-Segmentpassung gesteuert. Bereits während der Platzierung werden Rip-Up/Retry-Läufe eingeschoben, um die Ausnutzung der Platinenfläche zu optimieren.

Die Platzierungsoptimierung bietet Funktionen zum automatischen Bauteil- und Pin-/Gattertausch an. Der Bauteiltausch (Component Swap) führt eine iterative Vertauschung von Einbauplätzen bereits platzierter, identischer Gehäuse durch. Der Pin-/Gattertausch (Pin/Gate Swap) führt analog eine iterative Vertauschung von Gattern (auch bauteilübergreifend) bzw. Pins und Pingruppen durch. Die Zulässigkeit von Pin/Gate-Swaps wird anhand entsprechender Bibliotheksdefinitionen geprüft. Es lassen sich einzelne oder mehrere Optimierungsläufe aktivieren, wobei auch das Verfahren (nur Bauteiltausch, nur Pin-/Gattertausch, oder beide) entsprechend gewählt werden kann. Die Platzierungsoptimierung bewirkt in der Regel eine signifikante Vereinfachung der Verdrahtungsaufgabe und führt damit zu einer enormen Zeiteinsparung beim nachfolgenden Routvorgang.

Bartels Autorouter

Der **Bartels Autorouter**® ist das Produkt jahrelanger intensiver Forschung und praxisorientierter Erfahrung auf dem Gebiet der automatischen Leiterkartenentflechtung. Die Liste der Leistungsmerkmale des **Bartels Autorouters** ist beeindruckend. Der **Bartels Autorouter**® eignet sich zur automatischen Entflechtung von Leiterkarten die auf allen heutzutage gängigen modernen Leiterkartentechnologien (Mehrlagenlayouts, Analogdesigns, SMD-/SMT-Baugruppen, BGA-Komponenten, usw.) basieren. Die anwenderorientierte Konzeption, seine Zuverlässigkeit und seine Flexibilität haben den **Bartels Autorouter**® zum Industriestandard werden lassen. So wird der Router heute weltweit von großen CAD-Häusern unter den verschiedensten Namen und in unterschiedlichen Implementierungen angeboten. Im **Bartels AutoEngineer** ist selbstverständlich die jeweils aktuellste Version des **Bartels Autorouter®** mit allen Leistungsmerkmalen integriert. Durch den Einsatz des **Bartels Autorouter®** lässt sich der Zeitaufwand für den Entwurf von Leiterkarten drastisch reduzieren. Für den kompletten Entwurf einer Europakarte einschließlich Stromlaufplan und Fertigungsunterlagen benötigt ein erfahrener BAE-Anwender üblicherweise nicht länger als einen Tag.

Der **Bartels Autorouter**® besticht durch die Verknüpfung sehr hoher Entflechtungsintelligenz mit hervorragender Fertigungsqualität. Dieses Ziel wird durch einen speziellen Backtracking-/Ripup-/Reroute-Algorithmus erreicht, der die Leiterbahnen zunächst für eine vollständige Entflechtung optimal verlegt und dann das Layout noch einmal grundlegend in Richtung Fertigungsqualität optimiert. So werden bei der Entflechtung regelmäßig 100%-Ergebnisse erreicht. Der Algorithmus wird von einem Backtracking überwacht, das nicht nur eine Verschlechterung des Ergebnisses während Ripup oder Optimierung sowie ein Festfahren des Routers wirksam verhindert, sondern auch völlig neue Wegevarianten erschließt. Der *selektive* Ripup-/Retry-Algorithmus erlaubt es dem **Bartels Autorouter**® insbesondere auch, gezielt ganze Leiterbahnbündel zu verschieben, um so Platz für noch nicht verlegte Verbindungen zu schaffen (Push'n'Shove Routing).

Der netzübergreifende Optimierer gewährleistet durch das komplette Neuverlegen der Leiterbahnen nach Kriterien der Fertigungsfreundlichkeit eine hohe Layoutqualität. Dabei wird im Allgemeinen die Anzahl der Durchkontaktierungen (Vias) erheblich reduziert. Selbstverständlich werden auch die Leiterbahnecken abgeschrägt und Treppen - soweit sinnvoll - durch 45 Grad Leiterbahnen ersetzt.

Der **Bartels Autorouter**® kann bis zu 28 Lagen (16 Signallagen und 12 Versorgungslagen) simultan entflechten (Multilayer-Routing). Die Funktionen des **Autorouters** sind stark praxisorientiert. So kann z.B. gegen vorverlegte Kupferkämme gegengeroutet werden, um eine gute Stromversorgungsstruktur zu erreichen. Bei der Verwendung von Stromversorgungslagen werden selbstverständlich auch SMD-Pads richtig mit Durchkontaktierungen an diese angeschlossen. Generell wird beim Anschluss von SMD-Pads die vorgesehene Anschlussbreite nicht überschritten, um dem Aufstellen der Bauteile beim Löten vorzubeugen. Die Anschlussbreite ist pin- und nicht netzbezogen, T-Stücke werden automatisch konstruiert (Copper-Sharing).

Der Routingprozess kann am Bildschirm verfolgt werden. Die grafische Anzeige des Leiterkartenlayouts wird hierzu in Echtzeit aktualisiert, und es erfolgen statistische Anzeigen über den Routvorgang. Der Routingvorgang kann nach Bedarf abgebrochen, fortgeführt oder neu gestartet werden.

Bereits entflochtene Layouts können automatisch vom Router an veränderte Platzierungen oder Netzlisten angepasst werden (Änderungsrouten, Re-Entrant-Routing). Dabei entfernt der Router selbsttätig nunmehr falsche Leiterbahnen und ergänzt fehlende. Das modifizierte Layout kann dann nochmals optimiert werden.

Die heute in der Leiterplattentechnik eingesetzten Technologien werden vom Router voll unterstützt. Er erkennt alle frei definierbaren Strukturen (Lötaugen, Leiterbahnen, Kupferflächen) und schließt sie auf dem bestmöglichen Wege an. Auch nicht im Routingraster liegende Anschlüsse stellen durch die integrierte Offgrid-Erkennung kein Problem dar. Neben den Standard-Routingrastern mit wahlweiser Einstellung des Halbraster-Routingverfahrens (1/20 Zoll bis hin zu 1/100 bzw. 1/200 Zoll) stehen Optionen zur Vorgabe beliebiger Routingrasters für spezielle Pin-Grids und Optionen zum rasterfreien Routen zur Auswahl. Anschlüsse in geteilte Potentiallagen werden richtig erkannt und korrekt realisiert (Split Power Plane Routing). SMD-Technologien und BGAs (Ball Grid Arrays) werden durch spezielle Routing-Algorithmen zum Vorverlegen von SMD-Vias und zur Ankontaktierung von BGA-Anschlüssen unterstützt. Die Beherrschung partieller Durchkontaktierungen (Blind und Buried Vias) und Microvias (Via-in-Pad-Technologie) erhöht die Entflechtbarkeit von Multilayer-Platinen und unterstützt darüber hinaus neue Technologien der Leiterplattenfertigung wie z.B. Verfahren zur Realisierung plasmageätzter Durchkontaktierungen. Bereichsrouting ist mit Hilfe entsprechender Vorgaben für gesperrte Routingbereiche bzw. -Lagen möglich. Auch die Definition von Via-Sperrflächen ist möglich.

Im **Autorouter** stehen die aus dem **Layouteditor** bekannten Initialplacement-Funktionen sowie die Routinen zur Durchführung von Platzierungsoptimierungen zur Verfügung. Damit kann im **Autorouter** eine vollautomatische Vorplatzierung der Bauteile sowie eine Platzierungsoptimierung durch automatischen Bauteiltausch und Pin/Gate-Swap vorgenommen werden.

Darüber hinaus bietet der **Autorouter** eine Reihe spezieller Autorouting-Funktionen (Platzierungsoptimierung durch automatischen Pin/Gate-Swap im Rip-Up-Routing, rasterfreies Routen, Einzelnetz- und Bereichsrouting, Routen in gemischten Rastern, usw.) an. Die Funktion Platzieren/routen aktiviert sowohl den Voll-Autoplacer (mit komplettem Initialplacement und Platzierungsoptimierung) als auch anschließend den Voll-Autorouter (mit Initialrouting, Rip-Up/Retry-Routing und Optimierer), d.h. mit dieser Funktion lässt sich quasi auf Knopfdruck die gesamte Platzierung und Entflechtung vollautomatisch durchführen. Der Single-Net-Autorouter dient der automatischen Entflechtung einzelner, selektierbarer Signalnetze bzw. Verbindungen. Damit können z.B. Stromversorgungsnetze oder kritische Leiterbahnen unter Berücksichtigung spezifischer Vorgaben für Leiterbahnbreiten, Mindestabstände, Lagenzuordnungen, usw. selektiv vorverlegt werden. Der Vollständigkeit halber unterstützt der Single-Net-Router auch eine Option zum Löschen einzelner bereits gerouteter Netze bzw. Verbindungen. Das im **Autorouter** integrierte Verfahren zum Routen von Netzgruppen kann dazu verwendet werden, unterschiedliche, selektierbare Signalnetzgruppen (z.B. Busse eines speziellen Schaltungsblocks) selektiv mit spezifischen Routeroptionen (Vorzugsrichtungen, Leiterbreiten, etc.) zu verdrahten. Mit der Funktion zum Routen von Bauteilen (Component Routing) können Bauteile selektiv geroutet werden. Die Funktion zum Bereichs- oder Blockrouting dient der Entflechtung festlegbarer Bereiche bzw. selektierbarer Schaltungsblöcke auf der Leiterkarte. Damit können unterschiedliche Routingbereiche bzw. Schaltungsblöcke (I/O- oder Memorybereich, Digital- bzw. Analogbereich, etc.) mit unterschiedlichen, an die Schaltungstopologie angepassten Routeroptionen (Routingraster, Mindestabstände, Vorzugsrichtung, Vorgaben für Busrouting, etc.) entflochten werden. Während des Rip-Up-Autorouting-Prozesses bedient sich der **Autorouter** integrierter Funktionen zur Platzierungsoptimierung durch selektive Bauteil- und Pin/Gate-Swaps, um dadurch die Entflechtbarkeit der getauschten Bauteile, Gatter oder Pins zu erhöhen bzw. überhaupt zu ermöglichen.

Mit der im **Autorouter** integrierten Undo-Funktion können bis zu zwanzig Kommandos rückgängig und anschließend mit dem Redo-Kommando wieder ausgeführt werden. Dies gilt insbesondere auch für komplexeste Arbeitschritte wie z.B. komplette **Autorouter**-Läufe. Damit gewährleistet die Undo/Redo-Funktion einerseits Datensicherheit und ermöglicht andererseits eine komfortable Überprüfung von Realisierungsalternativen.

Durch die Einbindung der **Bartels User Language** in den **Autorouter** hat der Anwender die Möglichkeit, eigene Menüfunktionen (Makros), Report- und Testfunktionen, automatische Platzierungs- und Routingprozeduren, usw. zu implementieren, die er wahlweise explizit (über eine spezielle Menüfunktion) oder implizit (über Tastatur oder ereignisgeseteuert) aktivieren kann.

Die in **BAE HighEnd** integrierte Version des **Autorouters** bietet darüber hinaus mächtige Zusatzfunktionen basierend auf einer patentierten Technologie neuronaler Netzwerke (**Neuronaler Autorouter**). Der **Autorouter** der **BAE HighEnd**-

Software bedient sich künstlicher Intelligenz zur automatischen Lösung spezieller Entflechtungsprobleme wie sie z.B. typischerweise beim Analogrouting oder bei der Erzeugung von Leiterbahnen mit speziellen elektrischen Eigenschaften bzw. bei der Generierung von Mikrowellenstrukturen auftreten. Hierzu arbeitet der **Autorouter** mit Funktionen zur Erlernung und automatischen Anwendung von Regeln zur Lösung spezieller Entflechtungsprobleme und wird dabei zusätzlich noch unterstützt durch einen rasterlos arbeitenden, objektorientierten Routingalgorithmus mit integrierter Platzierungsoptimierung.

CAM-Prozessor

Da der **Bartels AutoEngineer** alle nur erdenklichen freien Geometrie-Definitionen zulässt, müssen auch die Postprozessoren diesen hohen Ansprüchen genügen. Natürlich hat der Pen- oder Fotoplotter nicht für jede Geometrie den passenden Stift oder die richtige Blende zur Verfügung. Deshalb werden die gewünschten Geometrien zunächst genau berechnet. Der intelligente Postprozessor wählt dann die bestmögliche Blende aus, um die vorgegebene Fläche exakt zu füllen. Nicht der Blendenteller definiert dabei die Lötaugen, sondern die geforderte Technologie. Die so generierten Unterlagen und Steuerdaten entsprechen genau den gewünschten Ergebnissen, sind mit den über den Bildschirm vorgegebenen Daten identisch. Damit hat der Anwender die freie Auswahl unter den Fotoplot-Dienstleistern und Leiterplatten-Herstellern, weil er alle benötigten Daten problemlos selbst erstellen kann.

Durch die Einbindung der **Bartels User Language** in den **CAM-Prozessor** hat der Anwender die Möglichkeit, eigene Menüfunktionen (Makros), Postprozessoren, CAM-Batchbetrieb, Report- und Testfunktionen, usw. zu implementieren, die er wahlweise explizit (über eine spezielle Menüfunktion) oder implizit (über Tastatur oder ereignisgesteuert) aktivieren kann.

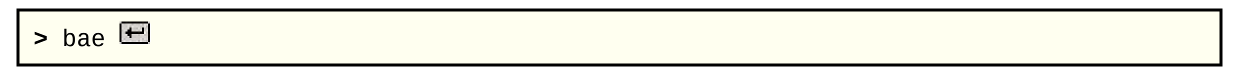
CAM-View

Das Modul **CAM-View** ermöglicht die visuelle Darstellung von Gerberdaten, Bohrdaten (Sieb&Meier bzw. Excellon) und Fräsdaten (Excellon) zur Kontrolle der Ausgabe und zur Überprüfung der verwendeten Werkzeuge. Darüber hinaus besteht mit **CAM-View** die Möglichkeit der Nutzengenerierung, und schließlich können mit Hilfe dieses Moduls Gerberdaten aus Fremdsystemen in den **AutoEngineer** übernommen werden.

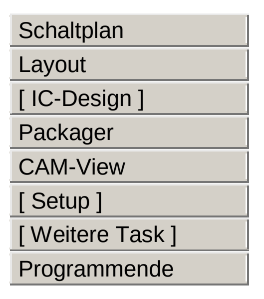
Durch die Einbindung der **Bartels User Language** in das **CAM-View**-Modul hat der Anwender die Möglichkeit, eigene Menüfunktionen (Makros), Batchprozeduren, Report- und Testfunktionen, usw. zu implementieren, die er wahlweise explizit (über eine spezielle Menüfunktion) oder implizit (über Tastatur oder ereignisgesteuert) aktivieren kann.

4.1.2 Programmaufruf

Der Aufruf des **Bartels AutoEngineer** sollte grundsätzlich aus dem Verzeichnis erfolgen, in welchem die zu bearbeitenden Projektdateien abgelegt bzw. abzulegen sind. Wechseln Sie also zunächst in Ihr Projektverzeichnis (zur Abarbeitung der in diesem Handbuch aufgeführten Beispiele ist es zweckmäßig, in das bei der Installation des **Bartels AutoEngineer** angelegte BAE-Jobs-Directory zu wechseln). Der Aufruf des Layoutmoduls erfolgt aus der Shell des **Bartels AutoEngineer**. Starten Sie diese von Betriebssystemebene aus mit folgendem Befehl:



Der **AutoEngineer** zeigt auf dem Schirm das Bartels-Logo sowie folgendes Menü (die Funktion Setup ist nur unter Windows bzw. Motif verfügbar; die Menüpunkte IC-Design und Weitere Task sind nur in speziellen Softwarekonfigurationen wie etwa in **BAE HighEnd** oder **BAE IC Design**) verfügbar):



Wählen Sie den Menüpunkt Layout mit der Maus an, und bestätigen Sie Ihre Wahl durch Drücken der linken Maustaste:



Nun wird der **Layouteditor** des **AutoEngineer** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

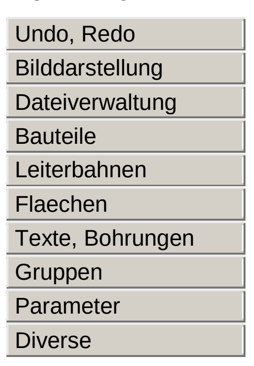
Der **Layouteditor** kann auch direkt aus dem **Packager** aufgerufen werden. In diesem Fall wird automatisch die Erzeugung eines Layoutelements für die zuletzt vom **Packager** erzeugte Netzliste vorgeschlagen, wenn noch kein entsprechendes Layoutelement existiert.

4.1.3 Hauptmenü

Nach dem Aufruf des **Layouteditors** befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden des **Layouteditors** ist das Menü Dateiverwaltung aktiviert, und der grüne Menübalken steht auf Laden.

Unter Windows und Motif kann anstelle der Standard- bzw. Seitenmenükonfiguration wahlweise auch ein Benutzerinterface mit Pulldownmenüs aktiviert werden. Hierzu ist mit Hilfe des Utilityprogramms **BSETUP** das Kommando `WINMENUMODE` mit der Option `PULLDOWN` in das Setup der BAE-Software einzuspielen (siehe hierzu auch Kapitel 7.2). Bei der Verwendung von Pulldownmenüs ist das Hauptmenü als horizontal ausgerichtete Menüleiste am oberen Ende der Benutzerschnittstelle angeordnet.

Das Hauptmenü ist während der Dauer der Layoutbearbeitung mit dem **Layouteditor** ständig verfügbar und ermöglicht die Aktivierung der folgenden Menüs:



Undo, Redo

Im Menü Undo, Redo finden Sie die Undo-Funktion, mit der die letzten zwanzig Arbeitsschritte rückgängig gemacht werden können. Mit der Redo-Funktion kann der Undo-Befehl wieder aufgehoben werden. Sie sollten diese wichtigen Funktionen unbedingt an einigen Stellen in den nachfolgenden Beispielen ausprobieren, um ein Gefühl für die Mächtigkeit dieser Kommandos zu bekommen.

Ansicht, Bilddarstellung

Im Menü Ansicht bzw. Bilddarstellung, das Sie außer durch Selektion im Hauptmenü auch immer über die mittlere Maustaste erreichen können, können Sie Zoomfunktionen aktivieren, das Eingabe- bzw. Hintergrundraster definieren, oder die Farbtabelle einstellen. Darüberhinaus finden Sie hier nützliche Hilfsfunktionen z.B. zur Bauteilsuche oder zur Elementabfrage.

Dateiverwaltung

Über das Menü Dateiverwaltung können Elemente neu generiert, geladen, gespeichert, kopiert, ersetzt oder gelöscht werden. Außerdem können von hier aus Farbtabellen geladen oder gespeichert werden, und es sind in diesem Menü auch wichtige Datenbank-Verwaltungsfunktionen (Auflisten Dateiinhalt, Update Bibliothek) enthalten.

Bauteile

Das Menü Bauteile enthält die Funktionen zur Definition von Platzierungs-Bauteilmengen, zur manuellen Bauteilplatzierung, zur Umbenennung von platzierten Bauteilen, zur automatischen Bauteilplatzierung (Matrixplacement, Initialplacement) sowie zur manuellen und automatischen Platzierungsoptimierung (Component Swap bzw. Pin/Gate Swap). Darüberhinaus kann in diesem Menü die Auswahl der Durchkontaktierung(en) für das Routing vorgenommen werden.

Auf Bauteilebene werden die Funktionen aus dem Menü Bauteile zur Definition bzw. Platzierung von Bauteilpins (Padstacks) verwendet. Auf Padstackebene werden die Funktionen aus dem Menü Bauteile zur Definition bzw. Platzierung von Pinanschlussflächen (Pads) verwendet.

Leiterbahnen

Im Menü Leiterbahnen sind die Funktionen zum interaktiven Verlegen von Leiterbahnen enthalten.

Flaechen

Das Menü Flaechen enthält die Funktionen zur Definition der Platinenumrandung, zur Erzeugung von Kupfer-, Potential- und Sperrflächen sowie zur Generierung von Dokumentarlinien oder Dokumentarflächen. Zur Bearbeitung bestehender Flächen stehen Funktionen zum Bewegen, Drehen, Spiegel, Kopieren und Löschen zur Verfügung. Darüberhinaus enthält das Menü Flaechen auch die Funktionen für die Flächenautomatik, d.h. zur Definition von Füllbereichen, zur automatischen Generierung und zum Löschen von Füll- und Schirmflächen sowie zur Erzeugung von Schraffurflächen.

Texte, Bohrungen

Das Menü Texte, Bohrungen dient dazu, Texte einzugeben, zu bewegen, zu verändern, oder wieder zu löschen. Daneben sind in diesem Menü die Funktionen zum Setzen und Löschen von Bohrungen enthalten.

Gruppen

Im Menü Gruppen werden Funktionen angeboten, mit deren Hilfe Teile des gesamten Layouts in Gruppen zusammengefasst und dann gespeichert, geladen, bewegt, gedreht, gespiegelt, skaliert, kopiert, gelöscht, fixiert oder freigegeben werden können.

Parameter

Das Menü Parameter enthält Funktionen zur Selektion der Bibliothek, zum Setzen des Nullpunktes bzw. der Elementgrenzen, zur Definition der Versorgungslagen, zur Selektion der Mincon-Funktion, zur Einstellung der Designregeln für die Abstandshaltung, sowie zur Aktivierung der automatischen Datensicherung.

Diverse

Im Menü Diverse kann der Programmabbruch oder der Rücksprung in die Shell des **Bartels AutoEngineer** veranlasst werden. Über dieses Menü des **Layouteditors** ist auch der Aufruf der Layoutmodule **Autorouter** und **CAM-Prozessor** möglich. Daneben werden hier weitere nützliche Funktion wie Batch Design Rule Check, Rueck-Netzliste und Flächen-Spiegelsicht angeboten. Auch der explizite Aufruf von **User Language**-Programmen ist von diesem Menü aus möglich.

4.1.4 Modifizierte Benutzeroberfläche

Menübelegung und Tastaturprogrammierung

Einige der mit der BAE-Software installierten **User Language**-Programme definieren implizite **User Language**-Programmaufrufe über die eine weit reichend modifizierte Benutzeroberfläche mit einer Vielzahl von Zusatzfunktionen (Startups, Toolbars, Menübelegung, Tastaturprogrammierung) aktiviert wird. Das **User Language**-Startupprogramm **BAE_ST** wird automatisch beim Aufruf des **Layouteditors** gestartet. **BAE_ST** ruft seinerseits das **User Language**-Programm **UIFSETUP** auf, welches eine vordefinierte Menü- und Tastaturbelegung im **Layouteditor** aktiviert. Änderungen bzw. Anpassungen der Menü- und Tastaturbelegung können *zentral* in der Quellcodedatei von **UIFSETUP** vorgenommen werden. Die aktuelle Tastaturbelegung kann mit dem **User Language**-Programm **HLPKEYS** angezeigt werden. Der Aufruf von **HLPKEYS** ist über die Funktion Tastaturbelegung aus dem Menü Hilfe möglich, sofern die vordefinierte Menübelegung aus **UIFSETUP** aktiviert ist. Mit dem **User Language**-Programm **UIFDUMP** kann die in der aktuellen Interpreterumgebung definierte Menü- und Tastaturbelegung in Form eines Reports angezeigt bzw. auf eine Datei ausgegeben werden. Mit dem **User Language**-Programm **UIFRESET** lässt sich die komplette Menü- und Tastaturbelegung zurücksetzen. **UIFSETUP**, **UIFDUMP** und **UIFRESET** sind auch über das Menü des **User Language**-Programms **KEYPROG** aufrufbar, welches zudem komfortable Funktionen zur Online-Tastaturprogrammierung sowie zur Verwaltung von Hilfstexten für **User Language**-Programme zur Verfügung stellt.

Kontextmenüs im Grafikarbeitsbereich

Bei Betätigung der linken Maustaste im Grafikarbeitsbereich wird ein kontextsensitives Menü mit spezifischen Funktionen zur Bearbeitung des an der aktuellen Mausposition platzierten Objekts aktiviert, wenn nicht bereits eine andere Menüfunktion aktiv ist. Ist kein Element geladen, dann werden die Dateiverwaltungsfunktionen Element laden bzw. Neues Element angeboten. Dieses Feature ist über einen automatisierten Aufruf des **User Language**-Programms **GED_MS** implementiert.

Kaskadierende Pulldownmenüs unter Windows/Motif

Die Windows- und Motifversionen des Grafikditors ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Pulldownmenüs der Windows- und Motifversionen des **Layouteditors** werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs ausgestattet. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

Dialoge für Parametereinstellungen unter Windows/Motif

In den Windows- und Motifversionen des **Layouteditors** sind die folgenden Dialoge für Parametereinstellungen implementiert:

- Einstellungen - Einstellungen: Allgemeine **Layouteditor**-Parameter
- Ansicht - Einstellungen: Bilddarstellungsparameter
- Bauteile - Autoplacement - Einstellungen: Parameter für automatische Platzierung
- Flächen - Flächenautomatik - Einstellungen: Flächenfüllparameter

In den Pulldownmenükonfigurationen werden die Standardfunktionen für Parametereinstellungen über das **User Language**-Programm **UIFSETUP** durch die obigen Menüfunktionen zum Aufruf der entsprechenden Dialoge ersetzt.

Pulldownmenükonfiguration unter Windows/Motif

Bei der Verwendung von Pulldownmenüs unter Windows und Motif wird über das **User Language**-Programm **UIFSETUP** eine an Windows angepasste Menüanordnung mit zum Teil geänderten Funktionsbezeichnungen und einer Vielzahl von Zusatzfunktionen konfiguriert. Das Hauptmenü des **Layouteditors** wird dabei wie folgt aufgebaut:



Das Menü Hilfe enthält die beiden Funktionen Referenzhandbuch und Hilfe zu. für den Zugriff auf das im Windows-Help-Format verfügbare Referenzhandbuch zum **Layouteditor**. Hilfe zu lädt dabei direkt die Referenzhandbuchseite eines selektierbaren Menüpunkts oder Benutzeroberflächenelements.

4.1.5 Grundsätzliches zur Bedienung

Automatische Parametersicherung

Im den Programm-Modulen des BAE-Layoutsystems sind Funktionen zur automatischen Sicherung wichtiger Design- und Bearbeitungsparameter implementiert.

Im **Layouteditor** werden bei Aktivierung der Funktion zur Sicherung des aktuell geladenen Elements folgende Parameter automatisch in der aktuell bearbeiteten Designdatei gespeichert:

- Zeitintervall für automatische Datensicherung
- Name der aktuell geladenen Layoutfarbtabelle
- Eingaberaster
- Hintergrundraster
- Raster- und Winkelfreigabe
- Koordinatenanzeigemodus
- Breitendarstellungswert
- Darstellungsmodus für Gruppenplatzierung
- Standardwinkel für Bauteilplatzierung
- Spiegelungsmodus für Bauteilplatzierung
- Airlineanzeigemodus für Bauteilplatzierung
- Standardtextgröße
- Standardleiterbahnbreiten
- Leiterbahnsegmentverschiebemodus
- Bibliothekszugriffspfad
- Mincon-Funktion
- Platzierungsmatrix
- Placement Matrix Enabled Flag
- Flächenautomatik Isolationsabstand
- Flächenautomatik Minimale Strukturgröße
- Flächenautomatik Leiterbahnaussparungsmodus
- Flächenautomatik Inselerkennungsmodus
- Flächenautomatik Wärmefallenerzeugung
- Flächenautomatik Wärmefallenstegbreite
- Schraffurlinienabstand
- Schraffurlinienbreite
- Schraffurmodus

Die Elementnamen der zu sichernden Parametersätze werden vom aktuell bearbeiteten Layoutelement abgeleitet. Layoutspezifische Parametersätze erhalten den Elementnamen des aktuell bearbeiteten Layouts, bauteilspezifische Parametersätze den Namen **`[part]`**, padstackspezifische Parametersätze den Namen **`[padstack]`**, padspezifische Parametersätze den Namen **`[pad]`**. Beim Laden eines Elements wird automatisch der entsprechende Parametersatz mitgeladen. Dadurch wird in komfortabler Weise eine spezifische Arbeitsumgebung zur Bearbeitung der selektierten Bibliothekshierarchie bzw. des selektierten Designobjekts aktiviert.

Lagenzuordnung

Gleichermaßen wichtig bei der Erstellung bzw. Anpassung von Bibliothekselementen wie beim eigentlichen Leiterkartendesign ist die Wahl der Lagenzuordnung. Das System stellt Ihnen folgende Lagen bzw. Anzeigeelemente zur Verfügung:

- Signallagen 1 - 100
- Signallage "Oberste Lage"
- Signallage "Alle Lagen"
- Signallage "Innenlagen"
- Versorgungslagen 1 - 12
- Dokumentarlagen 1 - 100
- Umrandung
- Unroutes
- Bohrungen
- Arbeitsbereich
- Nullpunkt
- Fehler
- Highlight
- Bohrung "-", "A" - "Z"
- Fixiert
- Verankerte

Die Signallagen unterliegen dem Design Rule Check. Auf ihnen wird das eigentliche Layout (Kupfer und Sperrflächen) dargestellt. Spezialfälle stellen hierbei die Signallagen Oberste Lage, Alle Lagen und Innenlagen dar.

Die Signallage Oberste Lage wird erst im Layout dynamisch einer bestimmten Signallage zugewiesen (die Standardeinstellung hierfür ist die Signallage 2). Diese Lage ist sehr nützlich, wenn die Lagenzahl oder Zuordnung beim Erstellen einer Bibliothek noch nicht festliegt. Die Zuweisung wird im Menü Einstellungen über die Funktion Oberste Lage durchgeführt und mit dem Layout gespeichert. Diese Funktion definiert auch die Position des Spiegels beim Spiegeln z.B. eines Bauteils. Es wird stets gegen die so festgelegte Lage gespiegelt, alle Lagen oberhalb der Obersten Lage bleiben unberührt.

Der Menütext zur Auswahl der obersten Lage kann über das BAE-Setup geändert werden kann (z.B. in Bauteilseite oder Component Side). Beachten Sie in diesem Zusammenhang die Möglichkeiten der Anpassung der Signallagenmenüs im Layout mit Hilfe des Utilityprogramms **BSETUP**.

Die Lage Alle Lagen wird als auf allen Signallagen (d.h. auf Signallage 1 bis einschließlich Oberste Lage) vorhanden betrachtet und dementsprechend geprüft und geplottet. Diese Lage ist z.B. für gebohrte Pins nützlich, wenn die Lagenzahl der Platine bei der Bibliothekserstellung noch nicht feststeht.

Die Lage Innenlagen wird als auf allen Signal-Innenlagen (d.h. auf allen Signallagen zwischen Signallage 1 und Oberste Lage) vorhanden betrachtet und dementsprechend geprüft und geplottet. Diese Lage vereinfacht für das Multilayer-Layout die Definition von Pins, die in den Signal-Innenlagen andere Anschlussformen besitzen als in den Signal-Außenlagen.

Die Versorgungslagen enthalten die sogenannten "Power Planes" und werden stets negativ dargestellt bzw. geplottet. Aussparungen und Wärmefallen werden durch den **CAM-Prozessor** automatisch erzeugt, die Festlegung der Signalnamen erfolgt im Menuü Einstellungen über die Funktion Versorgungslagen.

Die Dokumentarlagen dienen der Speicherung von Dokumentarinformation. Hier können Lötstopmasken, Klebemasken, Bestückungspläne, Bohrpläne, Passermarken, usw. definiert werden. Um das Erstellen von doppelseitigen SMD-Karten zu ermöglichen, ist jede Dokumentarlage in die folgenden Seiten unterteilt:



Es handelt sich hierbei prinzipiell um Unterlagen der jeweiligen Dokumentarlage. Beim Spiegeln werden stets Seite 1 und Seite 2 vertauscht, während beim Plotten wahlweise Beide Seiten zu Seite 1 bzw. Seite 2 hinzugefügt werden kann. Die Namen und Eigenschaften der Dokumentarlagen lassen sich individuell mit dem Programm **BSETUP** festlegen. Bei Änderungen mit **BSETUP** sollte jedoch beachtet werden, dass die Änderungen nur für neu definierte Grafikelemente wirksam werden. Dies ist notwendig, um einen Datenaustausch zwischen verschiedenen Installationen zu ermöglichen. Bitte befassen Sie sich daher vor der Erstellung von Bibliotheken genauestens mit der Lagenzuordnung und legen Sie die Ihren Anforderungen am besten entsprechende Zuordnung mit **BSETUP** vorher fest. Die Beschreibung des Utilityprogramms **BSETUP** sowie die mit der BAE-Software voreingestellten Dokumentarlagen-Definitionen finden Sie im Kapitel 7.2 dieses Handbuchs.

Die Lage Umrandung ist eine Speziallage, die lediglich der Definition der Platinenumrandung dient. Die Lage Unroutes ist eine Darstellungsebene für die Anzeige der noch nicht gerouteten Verbindungen (Unroutes, Airlines) auf dem aktuell geladenen Layout. Die Lage Bohrungen ist eine Darstellungsebene für die Anzeige der Bohrdefinitionen. Die Lagen Arbeitsbereich, Nullpunkt, Fehler und Highlight sind spezielle Ebenen zur Darstellung von Anzeigeelementen in der grafisch-interaktiven Benutzeroberfläche des BAE-Layoutsystems. Die Drill-Lagen dienen der Zuweisung von Farben an Bohrklassen. Die Speziallagen Fixiert und Verankert dienen der Auswahl von Mustern zur Anzeige fixierter bzw. verankerter Layoutelemente.

Vorzugslage und Elementauswahl

Häufig tritt der Fall ein, dass Elemente auf verschiedenen Lagen übereinander platziert sind (z.B. übereinander liegende Bauteile bei doppelseitig bestückten SMD-Platinen). Beim Pick eines Elements wird, sofern keine eindeutige Auswahl möglich ist, das auf der im Menü Ansicht mit der Funktion Vorzugslage selektierten Lage befindliche Element gewählt. Auch die erste beim Verlegen neuer Leiterbahnen angewählte Lage ist die Vorzugslage.

Die unter Einstellungen aus dem Menü Ansicht erreichbare Dialogbox enthält den Parameter Pickmodus zur Steuerung des Verhaltens bei Elementpicks an Positionen mit mehreren platzierten Elementen. Die Voreinstellung Vorzugslage bewirkt wie beschrieben den Pick eines Elementes anhand der Vorzugslage. Die Option Elementauswahl bietet bei mehreren Elementen an der Pickposition eine Elementauswahl an. In einer Schleife werden die Elemente nacheinander hervorgehoben dargestellt und eine Kurzbeschreibung in der Eingabezeile angezeigt. Durch Betätigung der Return-Taste oder der linken Maustaste kann das aktuell angezeigte Element selektiert werden. Mit der Escape-Taste oder der rechten Maustaste wird der Selektionsvorgang abgebrochen. Die Betätigung einer beliebigen anderen Taste bewirkt ein Weiterschalten zum nächsten Element an der Pickposition.

Netzliste

Die Netzliste stellt üblicherweise die verbindliche Vorgabe für den Layoutentwurf dar. Im **Bartels AutoEngineer** wird die Netzliste mit Hilfe des Schematic Editors generiert und durch den **Packager** in das Layout übertragen; daneben besteht die Möglichkeit, ASCII-Netzlisten mit Programmen wie **CONCONV** oder **REDASC** einzuspielen (siehe auch Kapitel 3 bzw. Kapitel 7).

Ein Layout wird grundsätzlich beim Laden dynamisch aufgebaut. Neben den auf dem Layout befindlichen Elementen aus den darunterliegenden Datenbank-Hierarchiestufen (Bauteile, Padstacks, Pads) wird auch die Netzliste mitgeladen und mit den Geometriedaten auf der Leiterkarte korreliert ("Connectivity Generierung"). Dies setzt allerdings voraus, dass der Layout Elementname und der Name der Netzliste identisch sind.

Nach erfolgreicher Connectivity-Generierung ist das System in der Lage, den Layoutentwurf ständig mit den Vorgaben in der Netzliste zu korrelieren. Dies stellt sicher, dass elektrische Verbindungen grundsätzlich erkannt werden, gleichgültig, ob sie über Leiterbahnen, Kupferflächen oder Durchkontaktierungen verlaufen. Verbindungen über Kreuzungen und T-Stücke sowie Aneinanderreihungen (z.B. bei Gruppenkopien) werden ebenfalls erkannt. Diese Funktionalität bezeichnet man als Copper Sharing oder Echte Connectivity.

Die Connectivity im **Layouteditor** der **BAE HighEnd**-Version baut auf modifizierten Datenstrukturen für die Kupferelemente auf. Diese enthalten mehr Information über die Struktur der elektrischen Verbindungen. Dies erfordert zwar einen erhöhten Speicherbedarf im Vergleich zu **BAE Professional**, bewirkt aber drastisch verringerte Rechenzeiten beim manuellen Bearbeiten großer Netze oder beim Verschieben von Bauteilen, da jeweils nur die direkt von der Änderung betroffenen Nachbargebiete in Echtzeit neu berechnet werden müssen.

Mincon-Funktion

Nicht realisierte Verbindungen werden als Luftlinien ("Airlines") dargestellt. Beim Platzieren von Bauteilen werden diese Airlines ohne Zeitverlust dynamisch zum jeweils nächst gelegenen Anschlusspunkt aktualisiert. Diese Funktion ist ein unverzichtbares Hilfsmittel, um zu einer optimalen Platzierung zu gelangen. Die Art der Luftliniendarstellung, d.h. die Methode zur Ermittlung des nächst gelegenen Anschlusspunktes wird mit der Mincon. Funktion im Menü Einstellungen ausgewählt.

In **BAE HighEnd** ist die Mincon-Berechnung auf Kosten eines im Vergleich zu **BAE Professional** erhöhten Speicherplatzbedarfs extrem beschleunigt.

Design Rule Check

Das System verfügt über einen Online-Design Rule Check. Dabei wird das Layout sowohl auf offene Verbindungen als auch auf Kurzschlüsse oder Abstandsverletzungen geprüft. Offene Verbindungen werden als Luftlinien angezeigt, Kurzschlüsse hell dargestellt (Highlight) und Abstandsverletzungen mit einem Rechteck eingerahmt. Die jeweiligen Farben können über die Farbpalette im Menü Ansicht eingestellt werden.

Der inkremental arbeitende Online-Check zeigt grundsätzlich nur alle *nach* dem Laden aufgetretenen Fehler an. Sofern auch alle vor dem Ladevorgang im Layout enthaltenen Fehler angezeigt werden sollen (z.B. vor der Generierung der Fertigungsdaten im **CAM-Prozessor**), ist nach dem Laden des Layouts die Funktion Batch-DRC (Menü Diverse) aufzurufen.

Die Mindestabstände für die Abstandsüberprüfungen werden mit den Abstandsfunktionen aus dem Menü Einstellungen bzw. wahlweise auch netzbezogen über die Netzliste (mit dem Netzattribut **MINDIST**) festgelegt. Die Abstände im Menü Einstellungen sind jedoch nicht unbedingt mit den Abständen im **Autorouter** identisch, da der **Autorouter** entsprechend der Routingraster-Vorgabe die Abstände selbsttätig festlegt. Netzbezogene Abstände hingegen werden auch vom Router individuell berücksichtigt.

Gruppen

Ein mächtiges Werkzeug stellen die Gruppenfunktionen im **Layouteditor** dar. Mit Hilfe dieser Funktionen können Teile des aktuell geladenen Layouts oder Bauteils zu Gruppen zusammengefasst und dann gespeichert, bewegt, gedreht, gespiegelt, skaliert, kopiert, gelöscht, fixiert oder freigegeben werden.

Die Gruppenfunktionen arbeiten nach dem Mengenprinzip. Es können Elemente wahlweise zur aktuell definierten Gruppe hinzugefügt (selektiert) oder auch wieder aus der Gruppe entfernt (deselektiert) werden. Die zur aktuell definierten Gruppe selektierten Elemente werden mittels Highlight angezeigt. Mit der Funktion Gruppe Polygon können mehrere in einem festzulegenden Polygonzug befindliche Objekte eines wählbaren Typs (Bauteile, Leiterbahnen, Flächen, Texte, nur sichtbare bzw. unsichtbare Elemente oder beliebige Elemente) selektiert bzw. deselektiert werden. Die Funktion Gruppe Einzelelemente dient der Selektion bzw. Deselektion von Einzelelementen (Bauteile, Leiterbahnen, Flächen, Texte). Hierbei besteht die Möglichkeit der repetitiven Objektauswahl, d.h. die Funktion bleibt mit den eingestellten Funktionsparametern solange aktiviert, bis kein gültiges Pickelement mehr angewählt wurde. Dadurch entfällt die sonst lästige Neuaktivierung der Auswahlfunktion bei der Selektion mehrerer Einzelelemente eines gewünschten Typs. Mit der Funktion Gruppe ruecksetzen können *alle* Elemente der aktuell definierten Gruppe deselektiert werden.

Alle zur aktuell definierten Gruppe selektierten Elemente werden in die nachfolgenden Gruppenfunktionen zum Speichern (Gruppe speichern), Bewegen (Gruppe bewegen), Kopieren (Gruppe kopieren), Löschen (Gruppe loeschen) einbezogen. Die Funktionen zum Fixieren (Gruppe fixieren) und Freigeben (Gruppe freigeben) wirken nur auf die zur Gruppe selektierten Bauteile und Leiterbahnen (inklusive Vias) des aktuell geladenen Layouts.

Die Ausführung der Funktionen Gruppe ruecksetzen, Gruppe Polygon, Gruppe loeschen, Gruppe fixieren und Gruppe freigeben wird durch eine Meldung über die Anzahl der geänderten Elemente quittiert. Offensichtliche Fehlselektionen lassen sich dadurch einfach erkennen.

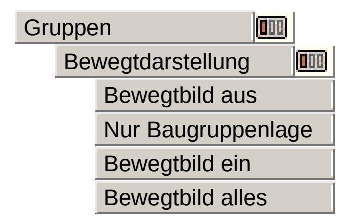
Mit der Funktion Gruppe speichern wird die aktuell definierte Gruppe als Dateielement abgespeichert. Hierbei ist ein Referenzpunkt zur Definition des Gruppenursprungs anzugeben. Das mit Gruppe speichern erzeugte Datenbankelement wird auf derselben Hierarchieebene angelegt wie das aktuell geladene Element. Um ein versehentliches Überschreiben existierender Datenbankelemente zu verhindern, aktiviert Gruppe speichern eine Bestätigungsabfrage für den Fall, dass ein Element mit dem spezifizierten Elementnamen bereits in der Zieldatei existiert. Mit Gruppe speichern lassen sich Teile eines erprobten Layouts oder Bauteils in Form von Templates zur späteren Wiederverwendung abspeichern. Solche Templates (wie übrigens auch beliebige Layouts oder Bauteile) können dann mit der Funktion Gruppe laden in andere Layouts bzw. Bauteile geladen werden.

Beim Kopieren und Laden von Gruppen mit Bauteilen (auf Layoutebene) oder von Gruppen mit Padstacks (auf Bauteilebene) ist zu beachten, dass die entsprechenden Gruppenfunktionen keine automatische Bauteil- bzw. Pinbenennung durchführen. Vielmehr werden bei möglichen Namenskonflikten die neu platzierten Bauteile bzw. Pins mit **#** benannt. Der Anwender muss dann im Bedarfsfall eine manuelle Umbenennung durchführen (ggf. kann dies auch automatisiert mit einem speziell hierfür implementierten **User Language**-Programm erfolgen).

Während des Bewegens von Gruppen mit einer der Funktionen Gruppe bewegen, Gruppe kopieren oder Gruppe laden kann mit der rechten Maustaste ein Untermenü mit Funktionen zur Platzierung auf Relativ- oder Absolutkoordinaten (Sprung relativ, Sprung absolut), zum Drehen bzw. Rotieren der Gruppe (Drehung links, Drehung rechts, Eingabe Winkel), zum Spiegeln der Gruppe (jeweils um die X-Achse mit Spiegelung ein bzw. Spiegelung aus) sowie zum Skalieren der Dimensionen und Platzierungskoordinaten der Gruppenelemente (Skalierung, durchgeführt nach Absetzen der bearbeiteten Gruppe) aktiviert werden. Im Untermenü der Funktion Gruppe bewegen steht nach Selektion des Verschiebestartpunktes zusätzlich die Option Quadrant setzen zur Verfügung. Nach Selektion dieses Menüpunkts sind der Ursprung für die Quadranteinteilung sowie der gewünschte Quadrant (Rechts oben, Links oben, Links unten oder Rechts unten) zu wählen. Die dann selektierte Verschiebung wird nur für Elemente bzw. Punkte vorgenommen die sich innerhalb des selektierten Quadranten befinden. Damit ist es möglich, Teile eines Layouts unter Beibehaltung der Verbindungen zum Rest des Layouts zu verschieben. Der hierbei verwendete Algorithmus ist allerdings kein echter Routingalgorithmus und lässt sich nur bei waagerechten bzw. senkrechten Verschiebungen sinnvoll anwenden. Bei komplexeren Verschiebungen ist ein Verwerfen der Leiterbahnen und ein erneuter **Autorouter**-Lauf vorzuziehen.

Die Funktion Gruppe laden setzt vor der eigentlichen Ladeoperation die aktuell definierte Gruppe zurück, d.h. es werden alle zum Zeitpunkt des Aufrufs der Funktion Gruppe laden selektierten Gruppenelemente deselektiert. Nachdem die Gruppe geladen ist, werden alle neu geladenen Gruppenelemente automatisch zur aktuellen Gruppe selektiert. D.h., die mit Gruppe laden geladenen Elemente (und nur diese) sind automatisch für die weitere Bearbeitung mit anderen Gruppenfunktionen selektiert.

Der **Layouteditor** unterstützt unterschiedliche Optionen zur Anzeige der zur Gruppe selektierten Objekte während des Platzierens der Gruppe. Hierfür stehen über die Funktion Bewegtdarstellung die folgenden Anzeigemodi für die Gruppen-Bewegtdarstellung zur Auswahl:



Mit Bewegtbild aus wird die Gruppen-Bewegtdarstellung abgeschaltet. Die Funktion Nur Baugruppenlage aktiviert die Anzeige der über das BAE-Setup mit dem Kommando **LAYGRPDISPLAY** des Utilityprogramms **BSETUP** (siehe hierzu Kapitel 7.2) definierten Dokumentarlage zur Gruppendarstellung; dabei erfolgt allerdings nur die Darstellung von Elementen der aktuell obersten Datenbank-Hierarchieebene. Die Option Bewegtbild ein aktiviert die Anzeige aller Gruppenelemente mit Ausnahme der Leiterbahnen, Durchkontaktierungen und Bohrungen. Bewegtbild alles aktiviert die Anzeige aller Gruppenelemente einschließlich der Leiterbahnen, Durchkontaktierungen und Bohrungen. Bewegtbild ein ist die Standardeinstellung.

Mit der Funktion Gruppe Macroname können zur Gruppe selektierte Bauteilmakros auf Layoutebene bzw. zur Gruppe selektierte Padstackmakros auf Bauteilebene ersetzt werden. Diese Funktion eignet sich insbesondere zum schnellen Austausch von Pintypen auf Bauteilebene. Auf Layoutebene werden mit Gruppe Macroname nur diejenigen Gehäusebauformen für in der Netzliste enthaltene Bauteile geändert, für die das angegebene Gehäuse als Alternativbauform definiert ist. Der Austausch von Gehäusebauformen für Bauteile, die nicht in der Netzliste enthalten sind, erfolgt hingegen ohne diese Prüfung.

Fixieren/Freigeben von Elementen

Bestimmte Elemente des Layouts wie Bauteile oder Leiterbahnen lassen sich fixieren. Fixierte Elemente werden in einem nachfolgend auszuführenden automatischen Prozess (Platzierungsoptimierung, Autorouting) nicht verändert. Wir empfehlen dies z.B. dringend für Bauteile, die vom Bauteil- bzw. Pin/Gate-Swap auszunehmen sind oder für den Stromversorgungskamm vor dem Start des **Autorouters**. Beim Start einer Fixierfunktion werden die zu diesem Zeitpunkt fixierten Elemente durch Highlight markiert, beim Verlassen der Funktion wird dieses spezifische Highlight wieder aufgehoben.

Beim Bearbeiten bzw. Kopieren fixierter Elemente (Bauteile, Leiterbahnen, Vias) bleibt die Fixierung erhalten bzw. wird auf die kopierten Elemente übertragen. D.h., die Funktionen zur manuellen Bearbeitung von Leiterbahnen im **Layouteditor** haben keine Einfluss auf aktuell gesetzte Fixiert-Attribute der bearbeiteten Leiterbahnelemente. Das bedeutet, dass nach einer manuellen Nachbearbeitung vorher fixierter Leiterbahnen keine Notwendigkeit einer neuerlichen Fixierung besteht, um in einem nachfolgenden **Autorouter**-Lauf die Umverlegung (oder gar Herausnahme) solcher vorverlegter Leiterbahnen zu unterdrücken.

Polygone, Flächen

Grundsätzlich kann jede Fläche einschließlich der Pinformen als Polygon eingegeben werden. Hierzu werden die im Menü Flächen vorhandenen Funktionen verwendet. Ein Polygon wird grundsätzlich einer Lage bzw. einem Flächentyp zugeordnet und darf beliebig viele Kreisbogenbestandteile enthalten. Bei der Erstellung von Kreisbögen wird zunächst die erste Bogenecke graphisch eingegeben und anschließend die Funktion Bogen links oder Bogen rechts im Untermenü angewählt. Danach erfolgt die Eingabe des Mittelpunktes, wobei der daraus resultierende Kreis dynamisch am Fadenkreuz angezeigt wird. Zuletzt wird dann der Endpunkt eingegeben, wobei der resultierende Kreisbogen dynamisch dargestellt wird.

Bei der Definition des Vollkreises entfällt die Eingabe des Kreisbogenendpunktes. Zunächst wählen Sie einen Eckpunkt an, selektieren anschließend Bogen links (oder Bogen rechts) und definieren den Mittelpunkt. Die Funktion ist dann unmittelbar nach Eingabe des Mittelpunktes anstelle der Eingabe des Endpunktes mit Fertig zu beenden.

Der Pickalgorithmus für Polygonbahnecken und Polygonsegmente sorgt dafür, dass bei mehreren möglichen Pickelementen im Fangbereich das Polygonelement mit dem minimalen Abstand zum Pickpunkt selektiert wird. Damit ist auch in Übersichtsdarstellungen ein gezieltes Anwählen von Polygonen möglich.

Attribute zur Projekt- und Versionskontrolle

Die Attribute **$pltfname**, **$pltfsname** und **$pltename** werden durch den Namen der Projektdatei, den Namen der Projektdatei ohne Pfadnamen bzw. den Namen des aktuell geladenen Elementes ersetzt.

Die Attribute **$plttime** (aktuelle Uhrzeit), **$pltdatede** (aktuelles Datum, deutsche Notation) und **$pltdateus** (aktuelles Datum, US-Notation) werden bei geladenem Layout bei der Bilddarstellung und bei der Plotausgabe jeweils durch die aktuelle Uhrzeit bzw. das aktuell Datum ersetzt. Die Attribute **$pltstime** (aktuelle Uhrzeit), **$pltsdatede** (aktuelles Datum, deutsche Notation) und **$pltsdateus** (aktuelles Datum, US-Notation) werden bei geladenem Layout bei der Bilddarstellung und bei der Plotausgabe jeweils durch die Uhrzeit bzw. das Datum der zuletzt für das aktuell geladene Layout durchgeführten Sicherung ersetzt. Dabei spielt es keine Rolle, auf welcher Datenbankebene (Pad, Padstack, Bauteil, Layout) der Attributtext definiert ist. Existiert ein gesetztes herkömmliches Attribut mit gleichem Namen für ein Symbol, so besitzt dieses Priorität bei der Anzeige bzw. Ausgabe.

Bemaßung

Mit dem Zeichen **#** als neuem Text wird eine Meßfunktion aktiviert, die die Distanz zweier danach abgefragter Punkte ermittelt und als Text darstellt. Die dabei zu verwendende Maßeinheit kann über die Funktion Koordinatenanzeige aus dem Menü Einstellungen zu Inch oder Millimeter festgelegt werden.

Berücksichtigung von Einschränkungen bei der CAM-Ausgabe

Alle Flächen, d.h. auch die Padformen können wahlfrei definiert werden. Bitte beachten Sie jedoch, dass der **CAM-Prozessor** bestimmte regelmäßige Formen wie Kreise und Rechtecke, sofern diese sich in den Grenzen der Symboltoleranz und Blendentabellen befinden, automatisch erkennt und wesentlich zeitsparender plotten kann. Dies beruht auf der allgemeinen Eigenschaft von Photoplottern, bestimmte Symbole blitzen zu können. Dadurch werden die Plotzeiten und Kosten erheblich reduziert. Grundsätzlich plottet der **CAM-Prozessor** jedoch alle Formen im Rahmen der gewählten Genauigkeit exakt und zeigt (durch Fehlermeldung und Highlight) an, wo dies aufgrund der Vorgaben nicht möglich ist. So werden z.B. auch alle Flächen beim Füllen automatisch an die Blendengröße bzw. Stiftbreite angepasst (automatischer Versatz um eine halbe Blende nach innen).

User Language

Im **Layouteditor** ist der **Bartels User Language Interpreter** integriert, d.h. vom **Layouteditor** aus können **User Language**-Programme gestartet werden. Der Anwender hat damit die Möglichkeit, eigene Zusatzfunktionen nach anwender- bzw. firmenspezifischen Bedürfnissen zu implementieren und in den **Layouteditor** einzubinden. Hierzu zählen zum Beispiel Statusanzeigen und Parametereinstellungen, Report- und Testfunktionen, Prüf- und Editierfunktionen, spezielle Plotfunktionen, Utilities zur Verwaltung von Bauteilbibliotheken, automatische Platzierungs- und Routingfunktionen, firmenspezifische Batch-Prozeduren, usw. usf.

Im **Layouteditor** können **User Language**-Programme explizit oder implizit aufgerufen werden. Der explizite Programmaufruf erfolgt über den Menüpunkt Anwenderfunktion im Menü Datei. Nach der Aktivierung dieses Menüpunktes ist auf die Abfrage nach dem Programmnamen der Name des aufzurufenden **User Language**-Programms (z.B. `ulprog`) explizit einzugeben. Die Betätigung einer beliebigen Maustaste oder die Eingabe eines Fragezeichens **?** auf die Abfrage nach dem Programmnamen bewirkt hierbei die Aktivierung eines Popupmenüs mit allen aktuell verfügbaren **User Language**-Programmen.

User Language-Programme können auch implizit über die Tastatur aktiviert werden. Diese Art des Programmaufrufs ist immer dann möglich, wenn nicht gerade eine andere interaktive Eingabe über Tastatur erwartet wird. Die Spezifikation des Programmnamens erfolgt dabei implizit durch Drücken einer Taste. Zulässige Tasten sind dabei die Standardtasten (1, 2, ..., 0, a, b, c, ...; entsprechende Programmnamen sind **ged_1**, **ged_2**, ..., **ged_0**, **ged_a**, **ged_b**, **ged_c**, ...) bzw. die Funktionstasten (F1, F2, ...; entsprechende Programmnamen sind dabei **ged_f1**, **ged_f2**, ...).

Der **Layouteditor** ermöglicht den ereignisgesteuerten Aufruf von **User Language**-Programmen. Dabei lösen spezielle Ereignisse bzw. Operationen implizit, d.h. automatisch den Aufruf von **User Language**-Programmen mit definierten Namen aus, sofern diese verfügbar sind. Im Einzelnen sind dies die **User Language**-Programme **GED_ST** beim Starten des **Layouteditors**, **GED_LOAD** nach dem Laden eines Elements, **GED_SAVE** vor dem Speichern eines Elements, **GED_TOOL** bei Selektion eines Toolbarelements sowie **GED_ZOOM** bei Änderung des Zoomfaktors. Der Aufruf über die Startupsequenz der Interpreterumgebung eignet sich besonders zur automatischen Voreinstellung von modulspezifischen Parametern sowie zur Tastaturprogrammierung und Menübelegung. Der implizite Aufruf von **User Language**-Programmen nach dem Laden bzw. vor dem Speichern von Elementen ermöglicht die automatische Aktivierung elementspezifischer Bearbeitungsparameter wie z.B. des zuletzt selektierten Zoombereichs oder spezieller Farbeinstellungen. Bei Interaktionen in der Werkzeugliste werden die den selektierten Toolbarelementen zugewiesenen Funktionen ausgelöst. Die Änderung des Zoomfaktors kann dazu benutzt werden, Aktualisierungen in Funktionen zur Verwaltung von Entwurfsansichten auszulösen.

Mit der **Bartels User Language** werden darüber hinaus mächtige Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Beachten Sie bitte, dass über die mit der BAE-Software ausgelieferten **User Language**-Programme eine Vielzahl von Zusatzfunktionen implementiert und transparent in die Benutzeroberfläche des **Layouteditors** eingebunden sind.

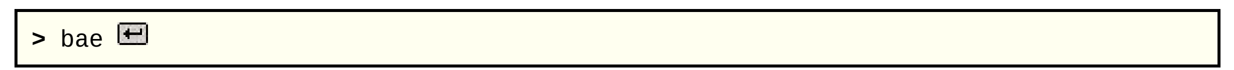
Eine ausführliche Beschreibung der **Bartels User Language** finden Sie im Bartels User Language Programmierhandbuch (Kapitel 4.2 enthält eine Auflistung aller mit der BAE-Software ausgelieferten **User Language**-Programme).

Neuronales Regelsystem

Im **Bartels AutoEngineer** sind eine Vielzahl mächtiger Zusatzfunktionen über das integrierte **Neuronale Regelsystem** implementiert. Kapitel 6.3.2 enthält eine Übersicht über die im Leiterkartenlayoutsystem bereitgestellten Regelsystemanwendungen.

4.2 Bibliotheksbearbeitung

Im Lieferumfang des **Bartels AutoEngineer** ist eine umfangreiche Bibliothek mit Layoutsymbolen enthalten. Natürlich kann jedoch der Fall eintreten, dass Sie für Ihr aktuell zu bearbeitendes Projekt Layoutsymbole benötigen, die noch nicht in der mitgelieferten Bibliothek definiert sind. Nachfolgend wird anhand von Beispielen die Erstellung derartiger Bibliothekssymbole beschrieben. Dabei werden entsprechend der Datenbankhierarchie (siehe hierzu auch Kapitel 1.3) ausgehend von der untersten Hierarchieebene zunächst verschiedene Pad- und Padstacksymbole und anschließend einige Gehäusesymbole (auf Bauteilebene) definiert. Alle diese Symbole werden in dem in den vorhergehenden Kapiteln bearbeiteten DDB-File **`demo.ddb`** abgelegt. Gehen Sie hierzu zunächst in das Verzeichnis, in dem **`demo.ddb`** abgelegt ist, und starten Sie den **AutoEngineer**:



Rufen Sie das Layoutmodul auf:



Sie befinden sich nun im **Layouteditor** des **Bartels AutoEngineer** und können mit der Erstellung der Bibliothekselemente beginnen. Bevor Sie jedoch (unter Verwendung der entsprechenden Datenblätter) eigene Layoutsymbole erzeugen, sollten Sie sich mit den Regeln zur Erstellung derartiger Symbole vertraut machen. Häufig existieren neben den Vorgaben aus den Datenblättern je nach zu verwendender Technologie firmen- bzw. fertigungsspezifische Konventionen, die zu beachten sind. Beispiele hierfür sind etwa Vorgaben für die Verwendung und Definition von Lötaugen, eine spezielle Kennzeichnung der Bauteil-Einbaulage bzw. des Pin 1 am Bauteil, das Raster zur Platzierung der Pins, der Bauteil-Nullpunkt für die Platzierung, der Bauteil-Pickpunkt für die automatische Bestückung, die Mindesthöhe für Texte, die Verwendung von Bohrsymbolen auf dem Bohrplan, einzuhaltende Bauteilmindestabstände (z.B. für SMDs auf der Lötseite), usw. In diesem Zusammenhang sollten Sie sich unbedingt auch mit der Lagenzuordnung (siehe Kapitel 4.1.5) vertraut machen, um eine korrekte Datenübergabe an die Fertigung zu ermöglichen.

Abbildung 4-1 zeigt die Layoutbibliothekssymbole, die wir in diesem Abschnitt erstellen werden.

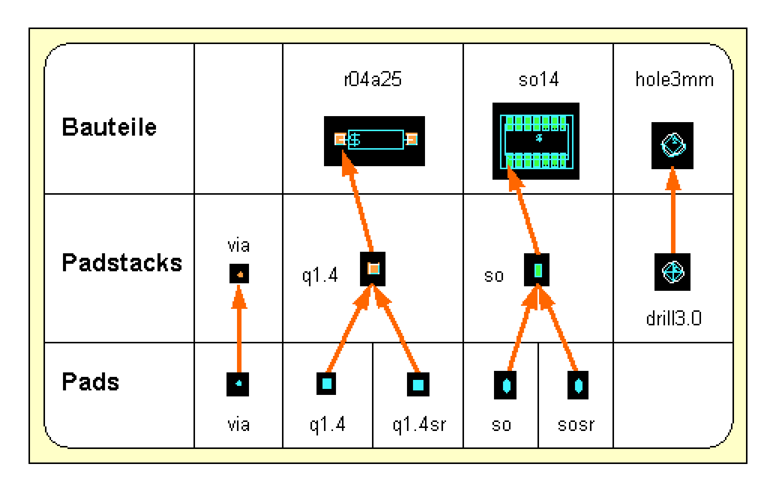


Abbildung 4-1: Layout-Bibliothekssymbole

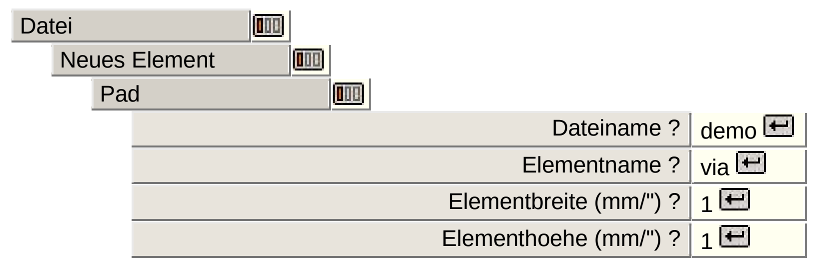
4.2.1 Paderstellung

Auf Padebene werden durch die Definition von Kupferflächen die Lötaugenformen festgelegt. Die in dieser Ebene generierten Padsymbole werden dann in der Padstackebene über die verschiedenen Lagen zu einem Stapel (Padstack) zusammengefasst und bilden die eigentliche Bauteil-Anschlussdefinition.

Ein Pad sollte in jedem Fall zunächst lagenfrei definiert werden (dies ist die Standardeinstellung im **BSETUP**) und erst im Padstack einer Signal- oder Dokumentarlage zugewiesen werden. Dadurch ist sichergestellt, dass bei einer eventuell notwendigen Änderung der Lagenzuordnung bei Verwendung einer anderen Fertigungstechnologie der Aufwand für die Bibliotheksanpassung möglichst gering ist.

Erzeugen eines Padsymbols

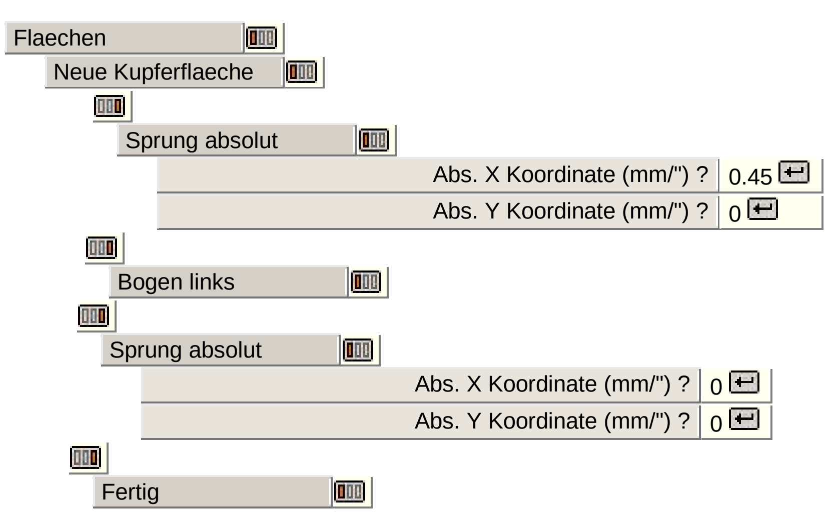
Generieren Sie mit den folgenden Befehlen in der Datei `demo.ddb` ein Padsymbol mit dem Elementnamen `via` und einer Elementgröße von 1*1mm:



Auf dem Bildschirm sehen Sie nun einen quadratischen Rahmen mit einem Kreuz in der Mitte. Der Rahmen beschreibt die Elementgrenzen des Pads, während das Kreuz die Position des Element-Nullpunktes kennzeichnet.

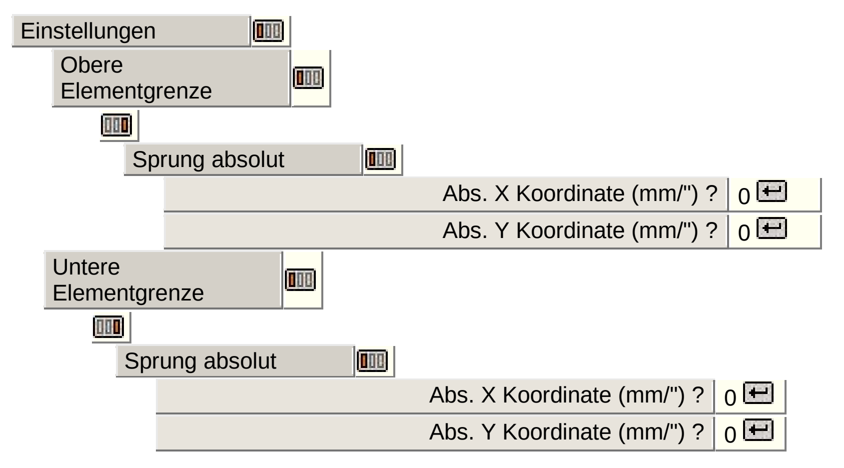
Definieren der Kupferfläche

Definieren Sie im Nullpunkt des Pads eine kreisförmige Kupferfläche mit einem Durchmesser von 0.9mm. Dies geschieht mit folgenden Befehlen:



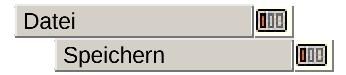
Definieren der Elementgrenzen

Sie können die Elementgrenzen des Padsymbols so umdefinieren, dass die Objekte des Pads möglichst dicht umschlossen werden. Dies geschieht wie folgt:



Sichern des erstellten Symbols

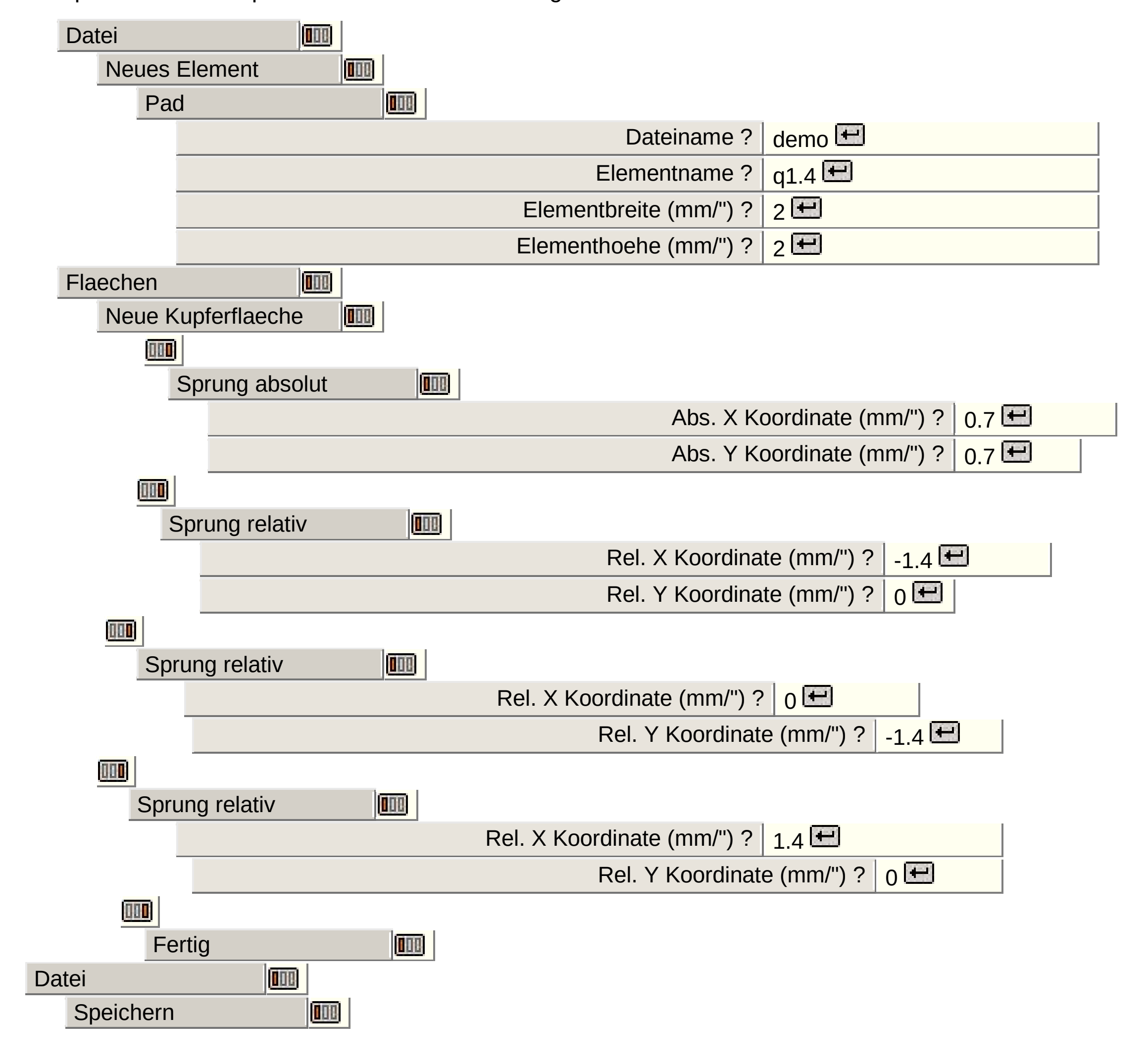
Sie haben theoretisch die Möglichkeit, am Padsymbol Dokumentation (Text, Dokumentarlinien oder Dokumentarflächen auf Dokumentarlagen) zu definieren. Da sich daraus aber eine sehr spezifische Paddefinition ergeben würde, die in recht wenigen übergeordneten Layoutsymbolen verwendet werden könnte, empfehlen wir, derartige Definitionen erst auf Padstack oder Bauteilebene vorzunehmen. Speichern Sie das Padsymbol nun mit folgenden Befehlen ab:



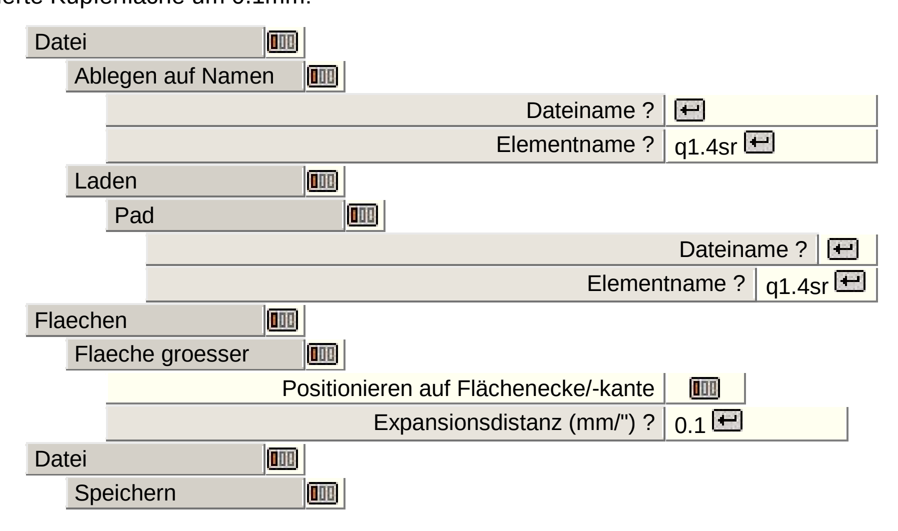
Das Padsymbol ist nun definiert und unter dem Elementnamen **`via`** in der Datei **`demo.ddb`** abgespeichert. Wir werden dieses Padsymbol später in einen Padstack laden, der für die Leiterbahn-Durchkontaktierungen (Umsteiger) beim manuellen bzw. automatischen Routen verwendet wird.

Definition quadratischer Pads

Definieren Sie nun mit den folgenden Kommandos in der Datei `demo.ddb` ein Padsymbol mit dem Elementnamen `q1.4` und einer quadratischen Kupferfläche mit der Kantenlänge 1.4mm:

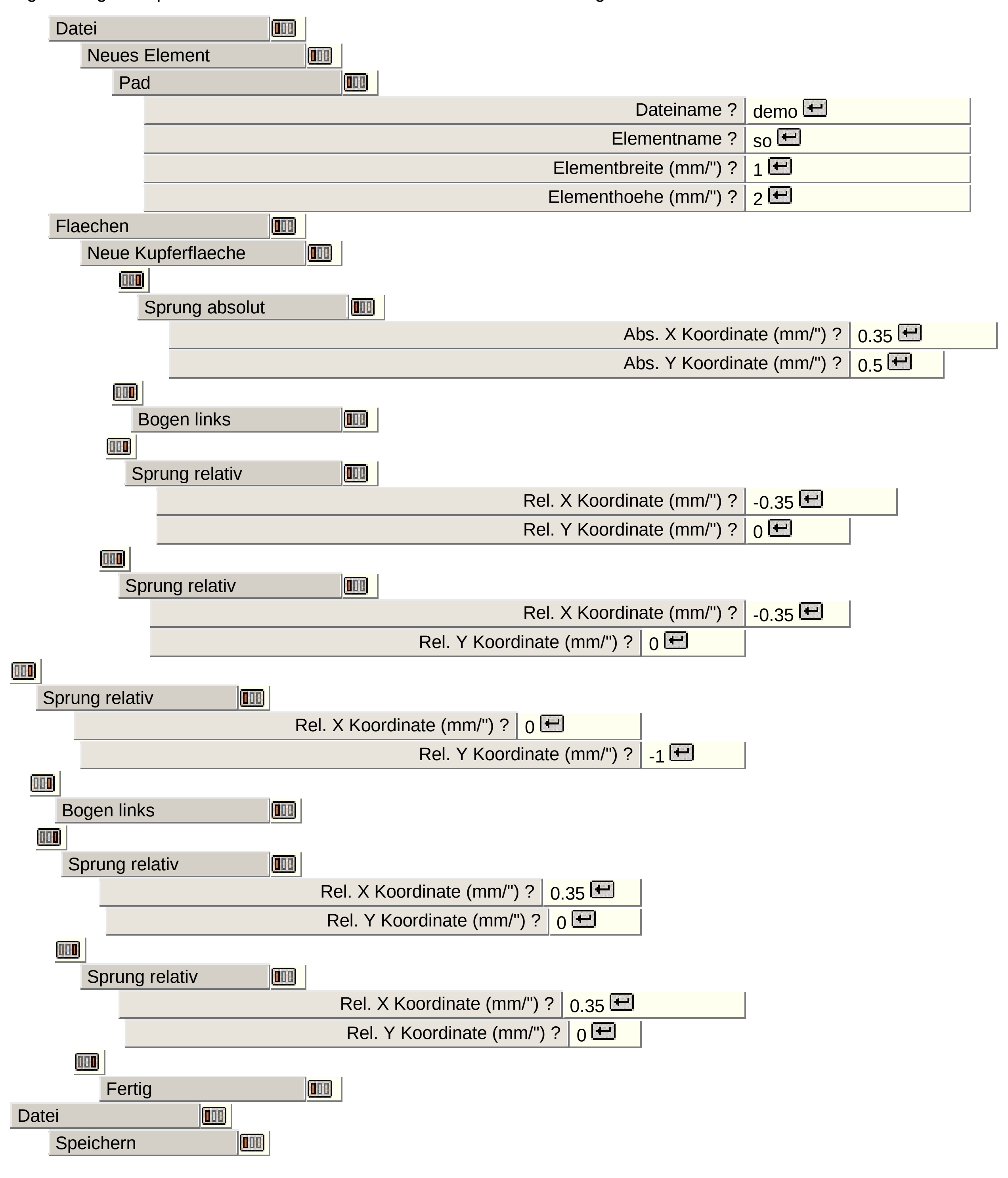


Im vorhergehenden Arbeitsschritt wurde das Pad `q1.4` definiert. Dieses Pad wird später in einem Padstack für bedrahtete Bauteile verwendet. Um auch ein dazugehöriges Symbol zur Generierung der Lötstopmaske zur Verfügung zu haben, ist ein entsprechendes Pad mit einer etwas größeren Kupferfläche zu erzeugen. Kopieren Sie mit den folgenden Kommandos das (noch geladene) Padsymbol `q1.4` auf `q1.4sr` (quadratisch 1.4mm, solder resist), und vergrößern Sie die auf `q1.4sr` definierte Kupferfläche um 0.1mm:

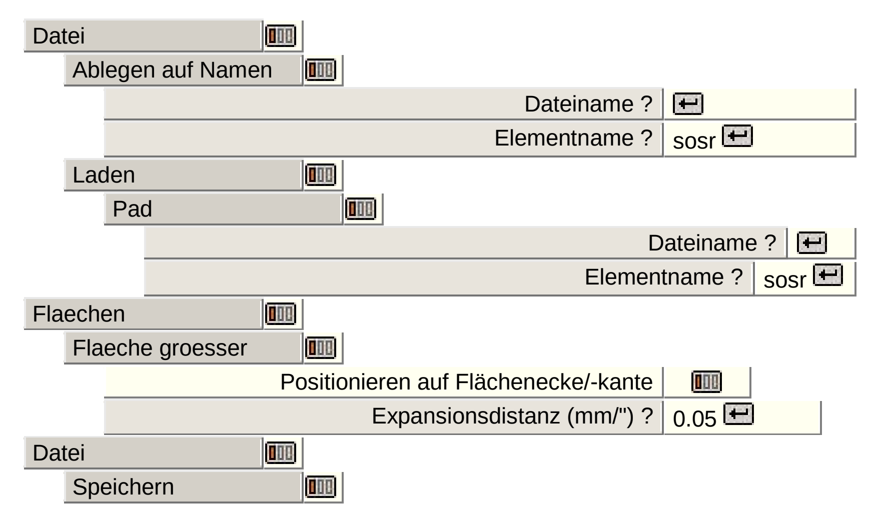


Definition von Finger-Pads

Definieren Sie nun mit den folgenden Kommandos in der Datei **demo.ddb** ein Padsymbol mit dem Elementnamen **so** und einer fingerförmigen Kupferfläche mit einer Breite von 0.7mm und einer Länge von 1.7mm:

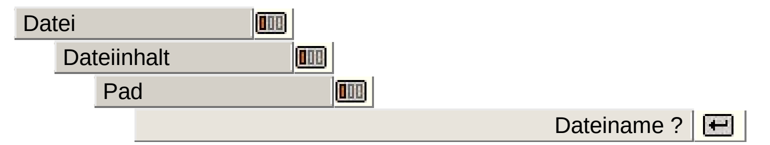


Im vorhergehenden Arbeitsschritt wurde das Pad **so** (SO-Gehäuse-Pad) definiert. Dieses Pad stellt die Lötaugenform für einen SMD-Anschluss dar. Generieren Sie mit den folgenden Kommandos aus dem (noch geladenen) Pad **so** das Pad **sosr** (für Lötstop, solder resist) mit einer um 0.05mm expandierten Kupferfläche:



Überprüfung des Dateiinhaltes

Sie haben nun eine Reihe von Padsymbolen definiert und in der Datei **demo.ddb** abgespeichert. Dies lässt sich wie folgt überprüfen:



Da bei der Abfrage nach dem Dateinamen ein leerer String angegeben wurde, verwendet das System den Dateinamen des im Speicher befindlichen Elements (also **demo.ddb**). Das System sollte nun im Grafikarbeitsbereich folgende Liste mit den in der Datei **demo.ddb** enthaltenen Padsymbolen ausgeben:

```
Typ : Pad / Datei : demo.ddb

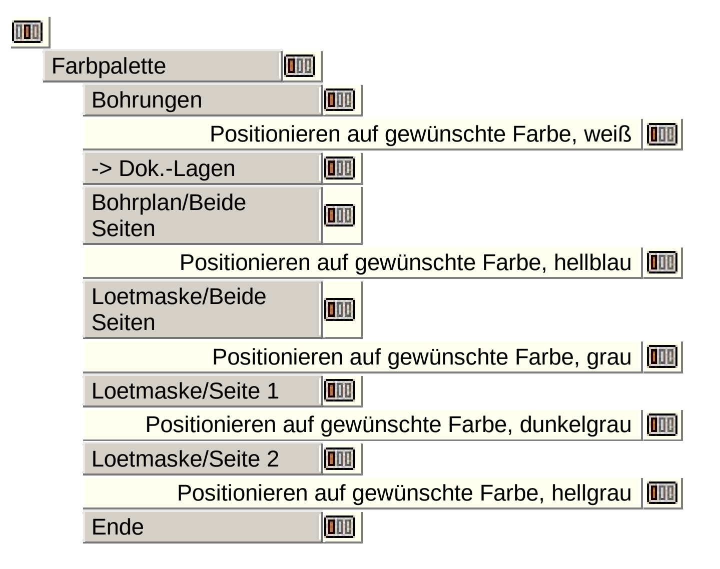
: q1.4            : q1.4sr          : so                : sosr
: via             - Ende -
```

Betätigen Sie nun die Leertaste, um wieder in die Menüoberfläche zu gelangen.

4.2.2 Padstackerstellung

Die in der Padebene erstellten Symbole werden auf Padstackebene über die verschiedenen Lagen zu einem Stapel (Padstack) zusammengefasst und bilden die eigentliche Bauteil-Anschlussdefinition. Im Padstack wird auch die Bohrinformation gespeichert, und zwar für gebohrte Pins in jedem Fall als Bohrungselement sowie wahlweise zusätzlich als Text und/oder Zeichnung auf der Bohrplan-Dokumentarlage. Für SMD-Anschlüsse entfällt beides. Auch können auf Padstackebene z.B. Sperrflächen definiert werden, um die Anschlussart des **Autorouters** zu bestimmen. Für reine Bohrlöcher (Befestigungsloch, Bezugsaufnahmeloch, ...) genügt die Definition eines Bohrungselements mit einer Sperrfläche, die verhindert, dass der **Autorouter** Leiterbahnen über das Bohrloch hinweg verlegt.

In den nachfolgend beschriebenen Arbeitschritten werden unter Verwendung der im vorherigen Abschnitt erzeugten Padsymbole einige Padstacks erstellt. Dabei werden Elemente auf möglicherweise über die Farbtabelle unsichtbar geschaltete Lagen abgelegt. Um eine Kontrolle über die jeweiligen Eingaben zu haben, empfiehlt es sich, die Farbtabelle für die Dauer der Padstackerstellung so umzudefinieren, dass die Bohrungen, die Bohrplan-Dokumentarlage Beide Seiten, sowie alle drei Seiten der Dokumentarlage Loetmaske sichtbar sind. Dies geschieht über das Menü Ansicht z.B. auf folgende Weise:



In den Farbauswahlmenüs erfolgt die Zuweisung einer Farbe an einen speziellen Anzeigeelementtyp durch Selektion des Anzeigeelements (bzw. der Lage) über die linke Maustaste sowie die anschließende Selektion der gewünschten Farbe. In den Farbauswahlmenüs des Layoutsystems besteht darüber hinaus die Möglichkeit der schnellen Lagen-Ein/Ausblendung mit Erhalt der aktuell eingestellten Farbe. Die Aktivierung bzw. Deaktivierung der Lagenanzeige erfolgt dabei durch Anwahl des Farbbuttons der gewünschten Lage mit der rechten Maustaste. In der Menüanzeige werden die Farbbuttons der aktuell ausgeblendeten Lagen durchgestrichen dargestellt.

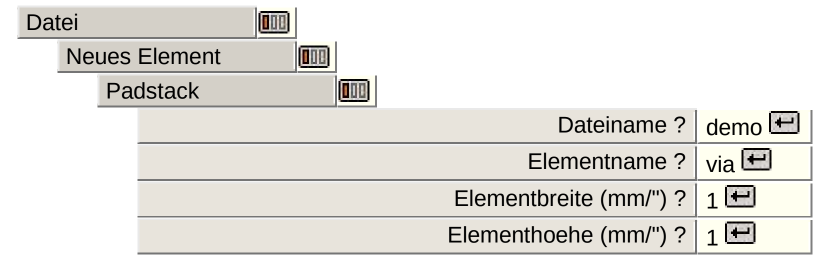
Die soeben definierte Farbtabelle können Sie mit den folgenden Kommandos unter dem Namen **`stackedit`** in der Datei **`ged.dat`** (im BAE-Programmverzeichnis) abspeichern:



Die so gespeicherte Farbtabelle kann später nach Bedarf mit der Funktion Farben laden aus dem Menü Ansicht wieder im Layoutsystem aktiviert werden.

Erzeugen eines Padstacksymbols

Generieren Sie mit den folgenden Befehlen in der Datei **demo.ddb** ein Padstacksymbol mit dem Elementnamen **via** und einer Elementgröße von 1*1mm:

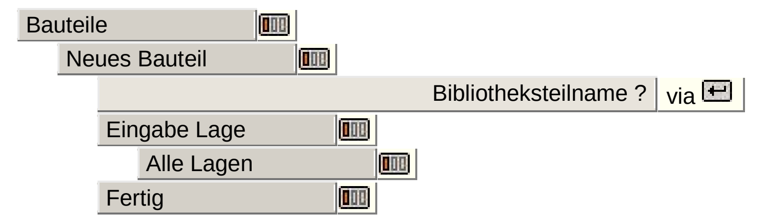


Auf dem Bildschirm sehen Sie nun einen quadratischen Rahmen mit einem Kreuz in der Mitte. Der Rahmen beschreibt die Elementgrenzen des Padstacks, während das Kreuz die Position des Element-Nullpunktes kennzeichnet.

Laden des/der Pads

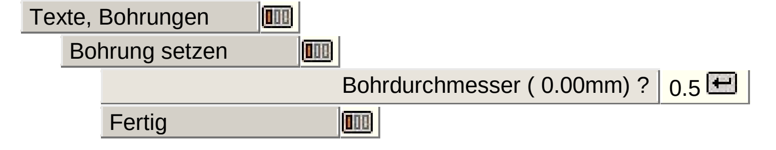
Mit der Funktion Neues Bauteil aus dem Menü Bauteile können Pads auf dem aktuell bearbeiteten Padstacksymbol geladen und platziert werden. Dabei erfolgt eine Abfrage nach dem Bibliotheksteilnamen, d.h. nach dem Namen des gewünschten Padsymbols. Hierbei wird zunächst ein Popupmenü zur Auswahl der Bibliotheksdatei angeboten, wobei die Dateinamensliste aus dem im BAE-Setup definierten Pfadnamen für die Layoutbibliothek abgeleitet wird, d.h. es werden alle im Verzeichnis der im System angemeldeten Standardlayoutbibliothek enthaltenen DDB-Dateien aufgelistet. Nach Auswahl der Layoutbibliothek (hier kann mit Projekt auch die aktuell bearbeitete DDB-Datei selektiert werden) erfolgt die Abfrage nach dem gewünschten Padsymbol (mit entsprechendem Popupmenü). Padsymbole können wahlweise auch direkt durch Eingabe des Bibliotheksdateinamens, eines Schrägstrichs (**/**) und des Elementnamens spezifiziert werden (ein Fragezeichen anstelle des Elementnamens aktiviert hierbei wiederum ein Popupmenü mit den in der angegebenen Bibliotheksdatei enthaltenen Padsymbolen). Durch die Eingabe eines Leerstrings auf die Abfrage nach dem Padsymbol wird das zuletzt mit Neues Bauteil selektierte Padsymbol ausgewählt (sofern zuvor bereits ein solches spezifiziert wurde).

Laden Sie mit den folgenden Kommandos das Padsymbol **via**, platzieren Sie es auf dem Padstack-Nullpunkt, und definieren Sie es als auf allen Signallagen vorhanden:



Definieren der Bohrung

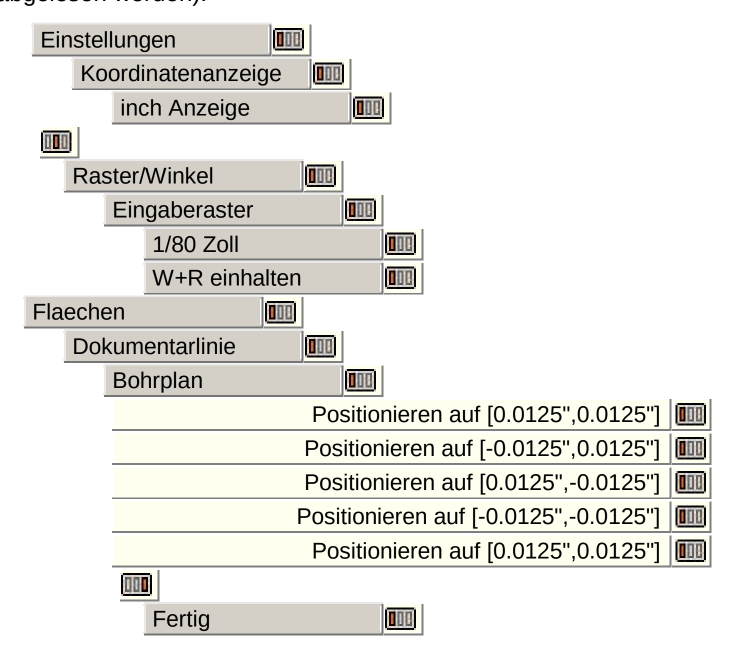
Definieren Sie mit den folgenden Kommandos eine Bohrung mit einem Durchmesser von 0.5mm:



Die im Untermenü der Funktion Bohrung setzen angebotene Option Bohrungsklasse ermöglicht die Zuweisung einer speziellen Bohrklasse für die Definition und Bearbeitung partieller Durchkontaktierungen (siehe hierzu auch Kapitel 4.6.11. Über die Option Spiegelbohr.klasse können Bohrungen wahlweise auch mit einer Spiegelungsbohrklasse versehen werden. Diese Bohrklasse wird aktiviert, wenn das Bauteil, auf dem sich die Bohrung befindet, gespiegelt wird. Damit ist die Definition spiegelbarer Bauteile mit partiellen Durchkontaktierungen möglich. Bei geladenem Padstack wird die optionale Spiegelungsbohrklasse unterhalb der Standardbohrungsklasse dargestellt. Auf Layoutebene ist immer nur die dem Spiegelungsmodus entsprechende Bohrklasse sichtbar.

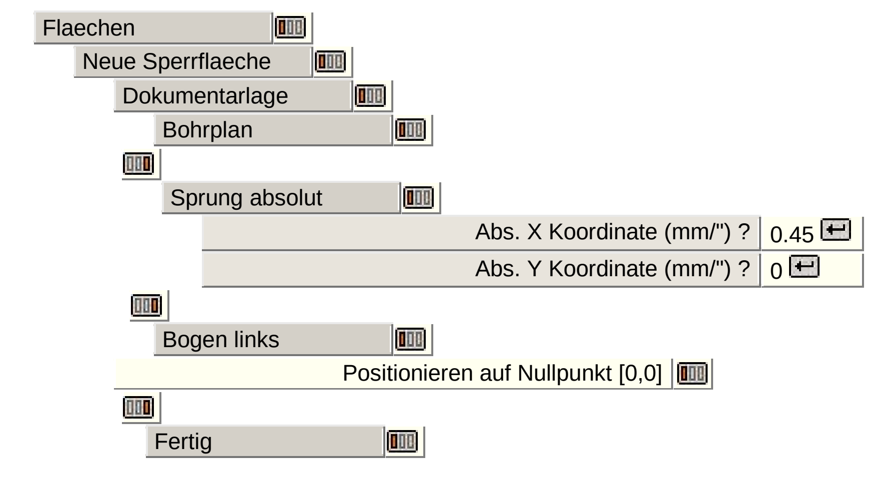
Definieren eines Bohrsymbols

Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bohrplan (Beide Seiten) zur Darstellung des Bohrplansymbols (es empfiehlt sich, vorher die Koordinatenanzeige auf Inch, und das Eingaberaster auf 1/80 Zoll umzustellen; die Koordinaten zur Positionierung auf die Linien-Eckpunkte können jeweils rechts oben im Info-Feld abgelesen werden):



Definieren einer Sperrfläche

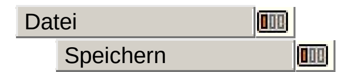
Definieren Sie mit den folgenden Kommandos entsprechend der Padform eine runde Sperrfläche mit einem Durchmesser von 0.9mm auf der Dokumentarlage Bohrplan (Beide Seiten):



Der Design Rule Check ist in der Lage, auf Dokumentarlagen definierte Sperrflächen gegeneinander zu prüfen. Obige Sperrflächendefinition bewirkt, dass der Design Rule Check Abstandsfehler meldet, wenn Durchkontaktierungen (beim manuellen Routen) versehentlich übereinander gesetzt werden.

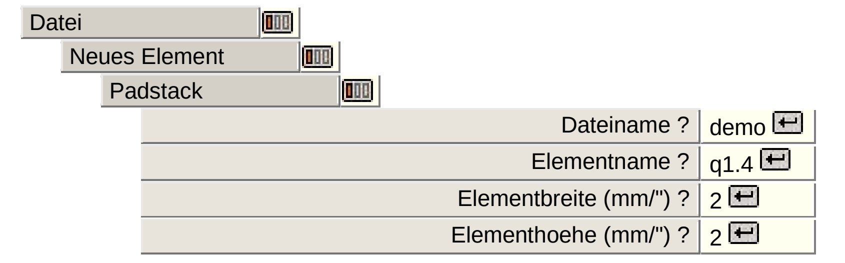
Sichern des erstellten Symbols

Das Padstacksymbol `via` ist nun definiert. Vergessen Sie nicht, das soeben erstellte *Symbol* zu *sichern*:

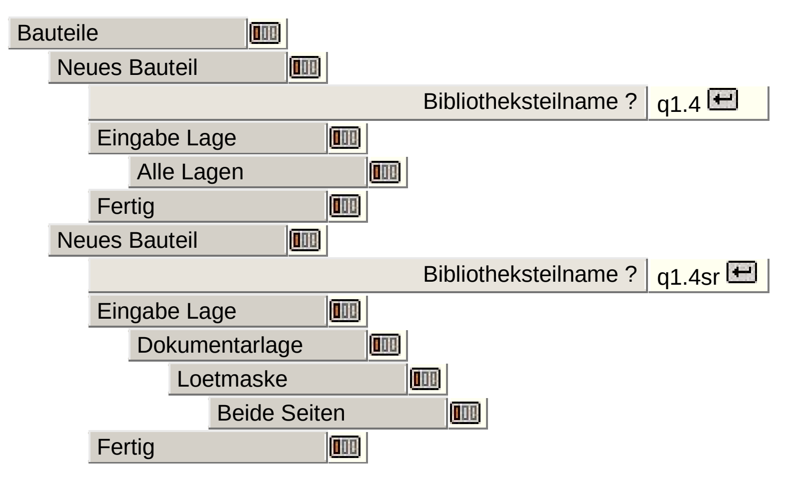


Definieren eines Padstacksymbols für gebohrte Bauteilanschlüsse

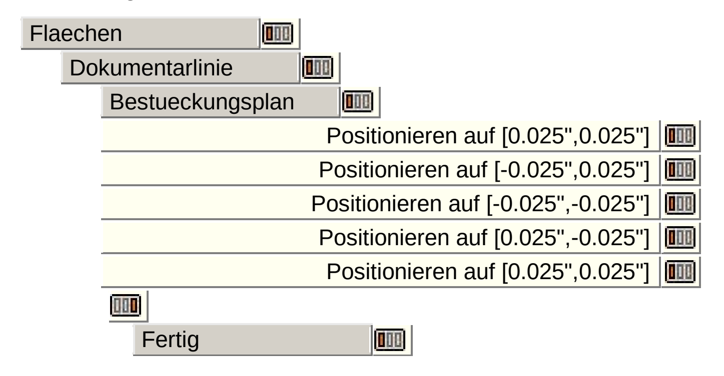
Erzeugen Sie mit den folgenden Kommandos in der Datei **`demo.ddb`** das Padstacksymbol **`q1.4`** mit einer Elementgröße von 2*2mm:



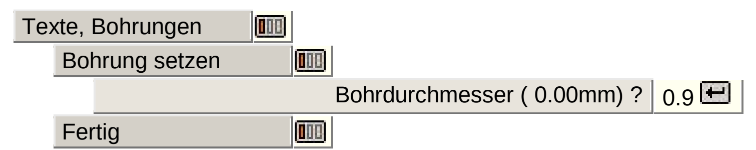
Laden Sie mit den folgenden Kommandos die Padsymbole **`q1.4`** (Alle Lagen) und **`q1.4sr`** (Dokumentarlage Loetmaske - Beide Seiten; für Lötstopmaske):



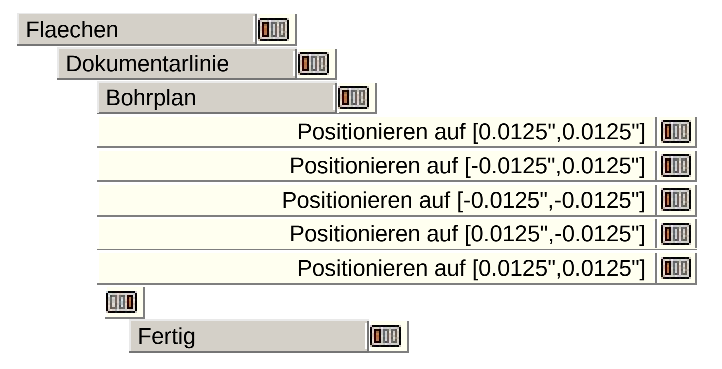
Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bestueckungsplan (Seite 2) zur Kennzeichnung des Anschlusses:



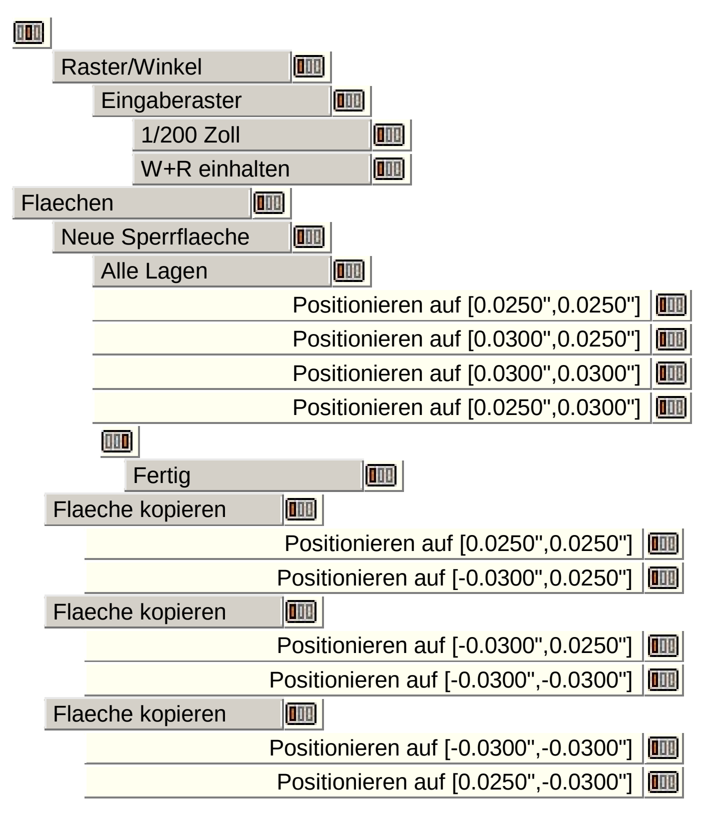
Definieren Sie eine Bohrung mit einem Durchmesser von 0.9mm:



Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bohrplan (Beide Seiten) zur Darstellung des Bohrplansymbols:

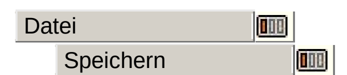


Stellen Sie das Eingaberaster auf 1/200 Zoll ein, und definieren Sie an den vier Eckpunkten des Padstacks jeweils eine kleine quadratische Sperrfläche auf Alle Lagen:



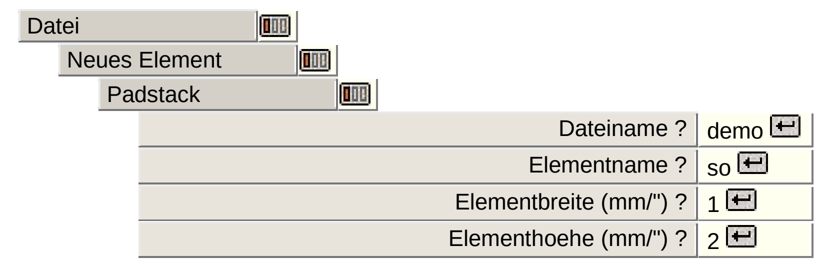
Mit Hilfe der in obigem Arbeitsgang definierten Sperrflächen wird die Anschlussart des **Autorouters** (nur orthogonal) für den Padstack `q1.4` festgelegt.

Vergessen Sie nicht, das soeben erstellte *Symbol* zu *speichern*:

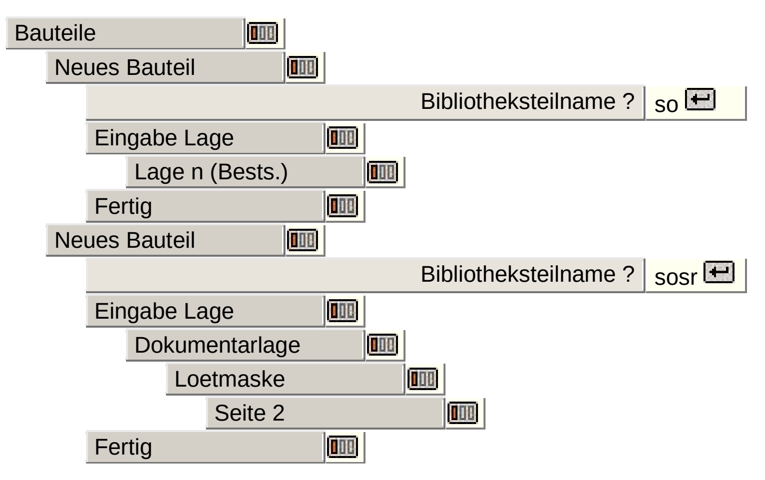


Definieren eines Padstacksymbols für SMD-Anschlüsse

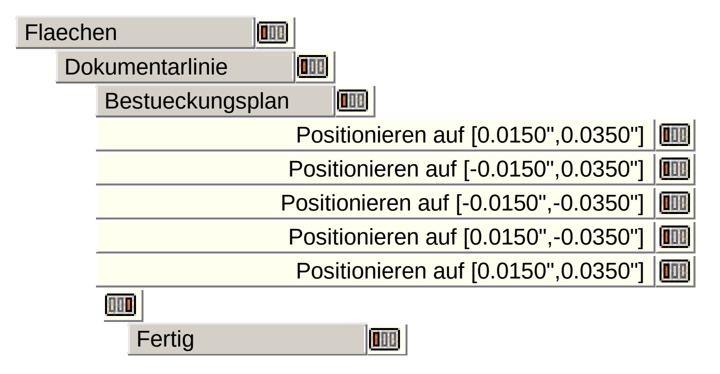
Erzeugen Sie mit den folgenden Kommandos in der Datei **`demo.ddb`** das Padstacksymbol **`so`** mit einer Elementgröße von 1*2mm:



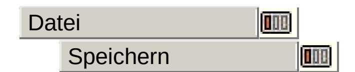
Laden Sie mit den folgenden Kommandos die Padsymbole **`so`** (Oberste Lage bzw. Bestueckseite) und **`sosr`** (Dokumentarlage Loetmaske - Seite 2; für Lötstopmaske):



Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bestueckunsplan (Seite 2) zur Kennzeichnung des Anschlusses:

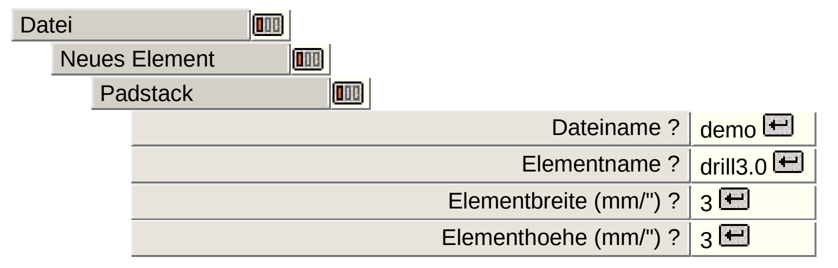


Vergessen Sie nicht, das soeben erstellte *Symbol* zu *speichern*:

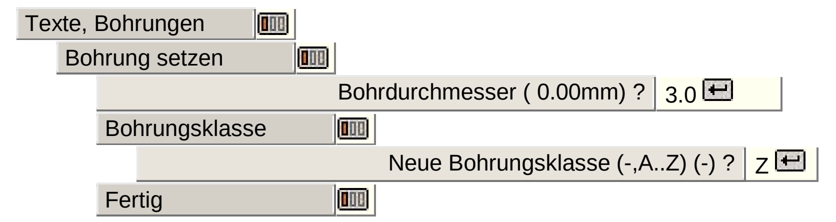


Definieren eines Padstack-Bohrsymbols

Erzeugen Sie mit den folgenden Kommandos in der Datei **`demo.ddb`** das Padstacksymbol **`drill3.0`** mit einer Elementgröße von 3*3mm:

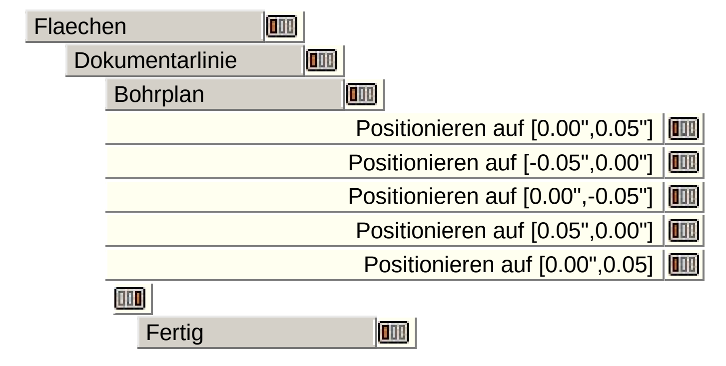


Definieren Sie mit den folgenden Kommandos eine Bohrung mit einem Durchmesser von 3.0mm, und ordnen Sie dieser Bohrung die Bohrungsklasse **`Z`** zu (dieses Kennzeichen kann später im **CAM-Prozessor** zur gezielten Ausgabe von nicht durchkontaktierten Bohrungen verwendet werden; siehe Kapitel 4.7.13, Ausgabe von Bohrdaten):

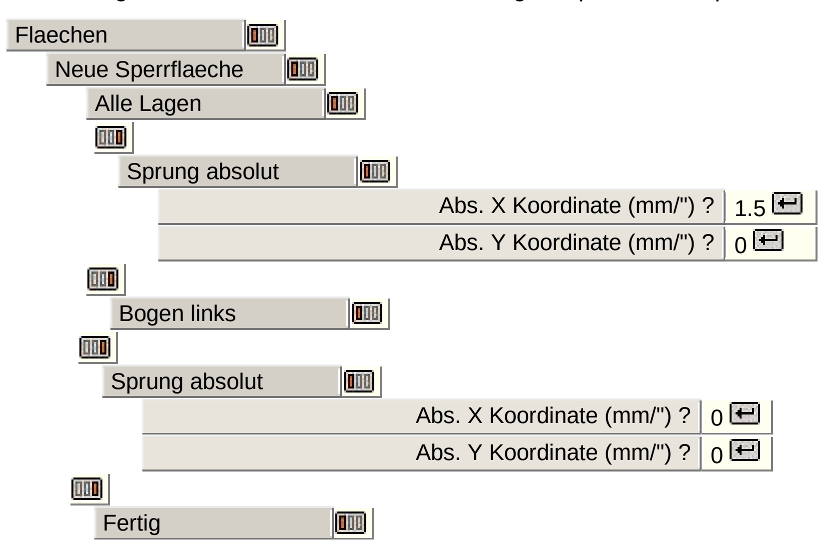


Die Eingabe eines Bindestrichs (**`-`**) auf die Abfrage nach der Bohrungsklasse bewirkt die Zuordnung der Bohrung zu keiner speziellen Bohrungsklasse. Dies ist zugleich die Standardeinstellung für die Definition von Bohrungen.

Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bohrplan (Beide Seiten) zur Darstellung des Bohrplansymbols:

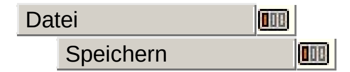


Definieren Sie mit folgenden Kommandos eine der Bohrung entsprechende Sperrfläche auf Alle Lagen:



Die Sperrflächendefinition ist notwendig, um zu verhindern, dass der **Autorouter** Leiterbahnen über die Bohrung legt.

Vergessen Sie nicht, das soeben erstellte *Symbol* zu *speichern*:



4.2.3 Bauteilerstellung

Auf der Bauteilebene werden die Gehäusebauformen erstellt. Die Bauteilanschlüsse werden durch das Laden und Platzieren von Padstacks definiert. Bauteile können auch Kupferflächen, Sperrflächen, Zeichnungsinformation, usw. enthalten. Der Bauteilname wird als Text durch das Attribut **$** definiert. Das **$**-Zeichen wird auf der nächst höheren Hierarchieebene, d.h. auf der Layoutebene durch den Bauteilnamen ersetzt. Auch die anderen Attribute (wie z.B. **$val**, **$llname**, usw.) können durch entsprechende Textdefinitionen auf Bauteilebene im Layout visualisiert werden.

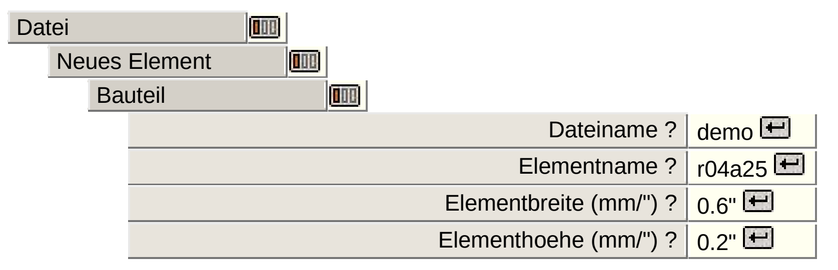
Es ist auch die Definition von Leiterbahnen und Durchkontaktierungen auf Bauteilebene möglich. Hierzu sind die im Kapitel 4.3.4 besprochenen Funktionen zum manuellen Routen anzuwenden. In der Layout-Connectivity und im Design Rule Check werden die Verbindungen bzw. Abstände zwischen Leiterbahnen, die auf demselben Bauteil definiert sind, nicht berücksichtigt. Dadurch ist es z.B. möglich, Bauteile zur verfahrenstechnisch korrekten Repräsentation gedruckter Spulen zu erzeugen, wobei die Pinverbindungen jeweils aus zwei Teilbahnen herzustellen sind. Kurzschlüsse bzw. Abstandsfehler, die durch andere Leiterbahnen auf Layoutebene mit den Leiterbahnen der gedruckten Spule entstehen, werden weiterhin erkannt und angezeigt. Da die Leiterbahnen ein und desselben auf Layoutebene platzierten Bauteils nicht mehr gegeneinander geprüft werden, wird dringend empfohlen, schon bei der Erstellung des Bauteils einen Batch-DRC auf Bauteilebene durchzuführen. Die Parameter für die Abstandsprüfung sind dabei auf die kleinsten für die Verwendung des Bauteils vorgesehenen Mindestabstände einzustellen.

Im Folgenden werden unter Verwendung der im vorhergehenden Abschnitt definierten Padstacks drei Bauteile definiert. Das erste Bauteil ist ein Widerstandsgehäuse mit gebohrten Anschlüssen; das zweite Bauteil ist ein 14poliges SMD-Gehäuse; das dritte Layoutsymbol ist ein konstruktives Element, das ein Bohrloch repräsentiert.

Bevor Sie mit der Erstellung der Bauteile beginnen, sollten Sie das Eingaberaster auf 1/20 Zoll (Menü Ansicht) und die Koordinatenanzeige auf Inch (Menü Einstellungen) einstellen.

Erzeugen eines Bauteilsymbols

Generieren Sie mit den folgenden Kommandos in der Datei **demo.ddb** ein Bauteil mit dem Elementnamen **r04a25** und einer Elementgröße von 0.6*0.2 Zoll:

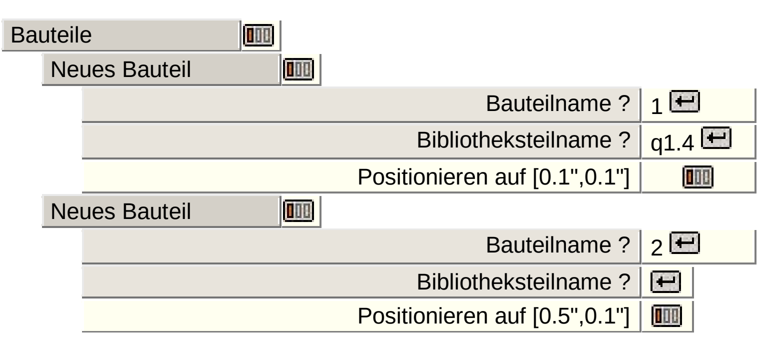


Auf dem Bildschirm sehen Sie einen rechteckigen Rahmen mit einem Kreuz links unten. Der Rahmen beschreibt die Elementgrenzen des Bauteils, während das Kreuz die Position des Element-Nullpunktes kennzeichnet.

Definieren der Bauteilanschlüsse

Mit der Funktion Neues Bauteil aus dem Menü Bauteile können Pins auf dem aktuell bearbeiteten Gehäusesymbol platziert werden. Dabei ist der Pinname und der Name des zu verwendenden Padstacksymbols zu spezifizieren. Beachten Sie, dass die in einer Bauteildefinition verwendeten Pinbezeichnungen eindeutig sein müssen. Bei Angabe einer bereits verwendeten Pinbezeichnung erfolgt daher eine Verifikationsabfrage, in der der Anwender den Ersatz des entsprechenden Pins ausdrücklich bestätigen muss. Bei der nachfolgenden Abfrage nach dem Bibliotheksteilnamen, d.h. nach dem Namen des gewünschten Padstacksymbols wird zunächst ein Popupmenü zur Auswahl der Bibliotheksdatei angeboten, wobei die Dateinamensliste aus dem im BAE-Setup definierten Pfadnamen für die Layoutbibliothek abgeleitet wird, d.h. es werden alle im Verzeichnis der im System angemeldeten Standardlayoutbibliothek enthaltenen DDB-Dateien aufgelistet. Nach Auswahl der Layoutbibliothek (hier kann mit Projekt auch die aktuell bearbeitete DDB-Datei selektiert werden) erfolgt die Abfrage nach dem gewünschten Padstacksymbol (mit entsprechendem Popupmenü). Padstacksymbole können wahlweise auch direkt durch Eingabe des Bibliotheksdateinamens, eines Schrägstrichs (`/`) und des Elementnamens spezifiziert werden (ein Fragezeichen anstelle des Elementnamens aktiviert hierbei wiederum ein Popupmenü mit den in der angegebenen Bibliotheksdatei enthaltenen Padstacksymbolen). Durch die Eingabe eines Leerstrings auf die Abfrage nach dem Padstacksymbol wird das zuletzt mit Neues Bauteil selektierte Padstacksymbol ausgewählt (sofern zuvor bereits ein solches spezifiziert wurde).

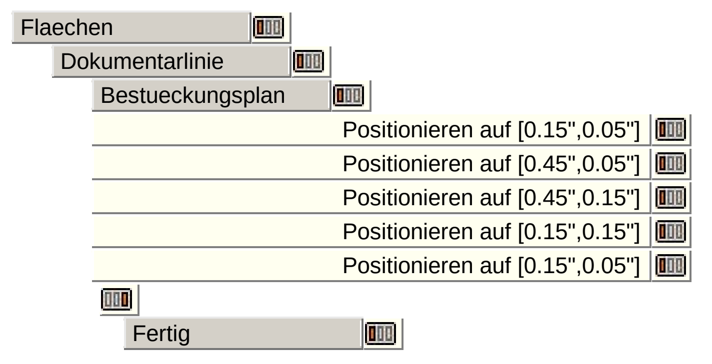
Definieren Sie die Anschlüsse **`1`** und **`2`** des Bauteils durch Laden und Platzieren des in Kapitel 4.2.2 erstellten Padstacksymbols **`q1.4`**:



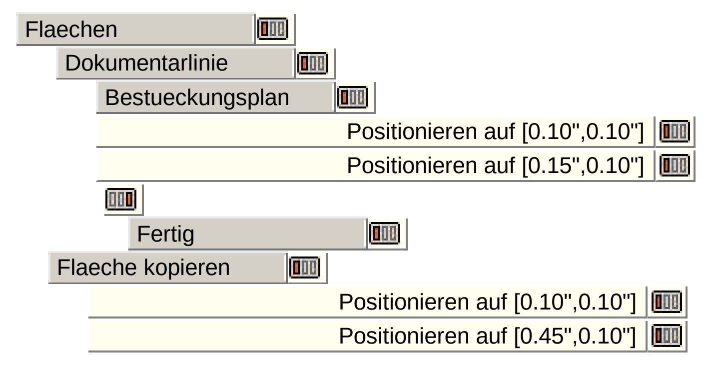
Wird bei der Abfrage nach dem Bibliotheksteilnamen für den zu ladenden Padstack ein Leerstring eingegeben (Betätigen der Eingabetaste ↵), dann verwendet das System das zuvor geladene Padstacksymbol. Durch die Definition der Bauteilanschlüsse ist das Bauteil von der Logik her bereits vollständig beschrieben. Was nun noch fehlt, sind z.B. Zeichnungsinformationen, die für die CAM-Ausgabe benötigt werden.

Zeichnungsinformation für den Bestückungsplan

Definieren Sie mit den folgenden Kommandos eine Dokumentarlinie auf der Dokumentarlage Bestueckungsplan (Seite 2) zur Kennzeichnung des Bauteilumrisses:

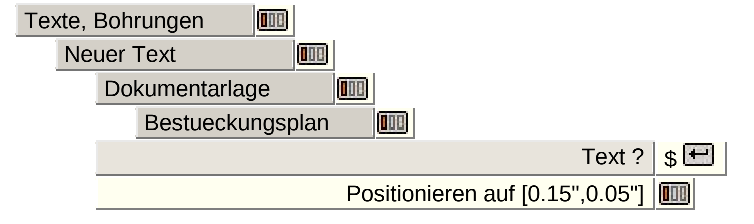


Definieren Sie nun mit den folgenden Kommandos zwei Linien vom Bauteilumriss zu den Bauteilanschlüssen auf der Dokumentarlage Bestueckungsplan:



Platzhalter für den Bauteilnamen

Tragen Sie mit den folgenden Kommandos auf der Dokumentarlage Bestueckungsplan (Seite 2) den Text **$** ein:



Anstelle des **$**-Zeichens erscheint auf Layoutebene der Bauteilname. Derartige Textdefinitionen können später für die Bestückdatenausgabe verwendet werden. Bei der Bestückdatenausgabe werden anstelle eines Plots die auf der betreffenden Lage definierten Texte mit Koordinate und Drehwinkel ausgegeben. Aus diesem Grund ist dafür zu sorgen, dass der Text exakt auf die Position des Pickpunkts für den Bestückautomaten gesetzt wird. Zusätzlich darf die Position des Textes für die Bestückdatenausgabe auf Layoutebene nicht versehentlich (durch die Funktion Name bewegen im Menü Texte, Bohrungen) verändert werden. Dies kann durch eine entsprechende Definition der betreffenden Bestückdaten-Dokumentarlage mit Hilfe des Programms **BSETUP** (siehe Kapitel 7.2) sichergestellt werden.

Definition des Nullpunkts

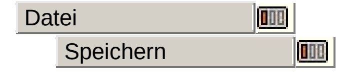
Setzen Sie mit den folgenden Kommandos den Nullpunkt des Bauteils auf den Pin **1**:



Der Nullpunkt des Bauteils ist auch der Referenzpunkt für die spätere Platzierung auf dem Layout. Bei Gehäusen mit gebohrten Anschlüssen ist dies meist die Position des ersten Pins, während bei SMD-Bauteilen häufig der Schwerpunkt des Symbols als Nullpunkt zu definieren ist.

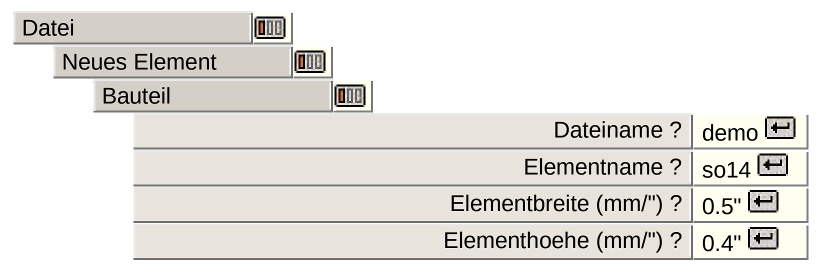
Sichern des erstellten Symbols

Das Bauteilsymbol **r04a25** ist nun vollständig definiert. Vergessen Sie nicht, dieses *Symbol* zu *sichern*:

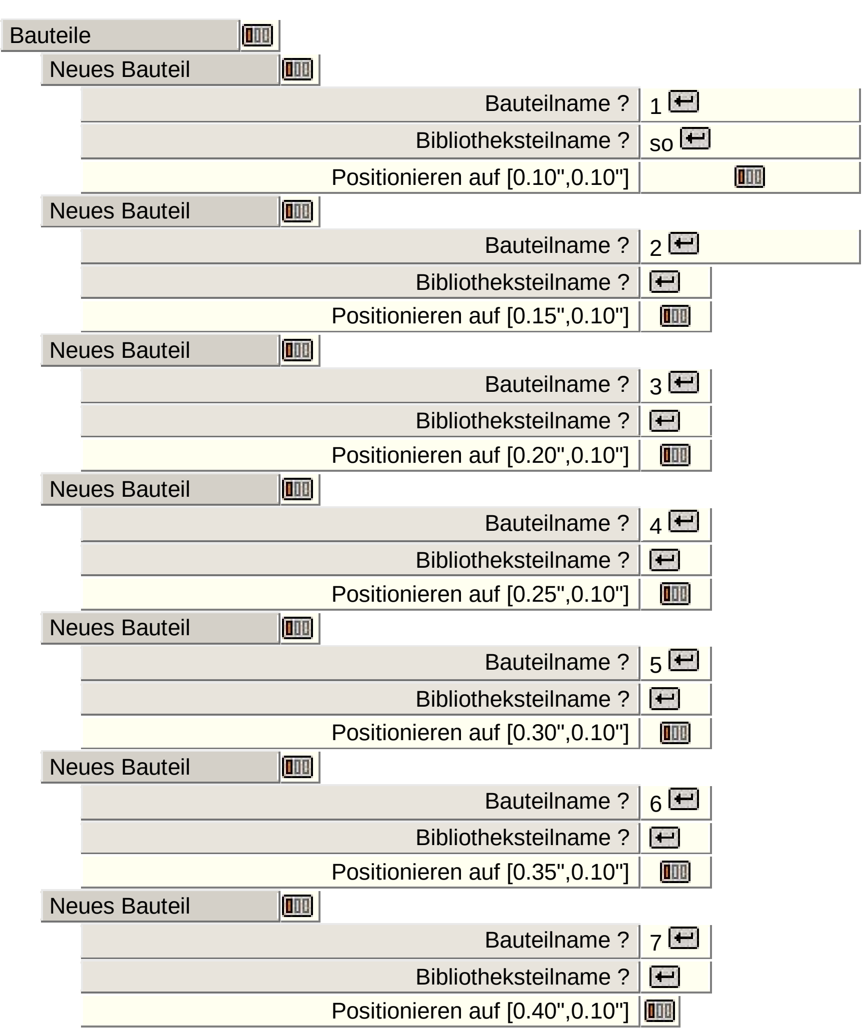


Erstellung eines SMD-Bauteils

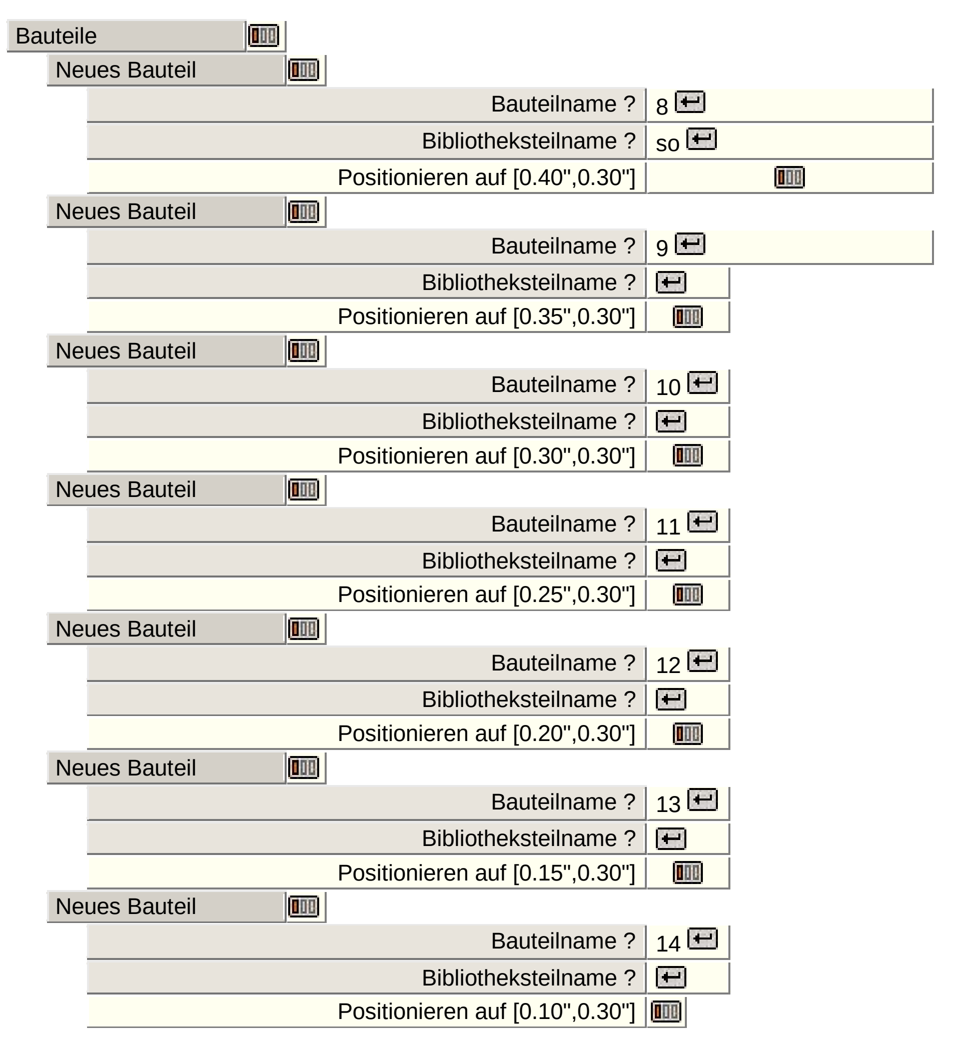
Generieren Sie mit den folgenden Kommandos in der Datei **demo.ddb** ein Bauteil mit dem Elementnamen **so14** und einer Elementgröße von 0.5*0.4 Zoll:



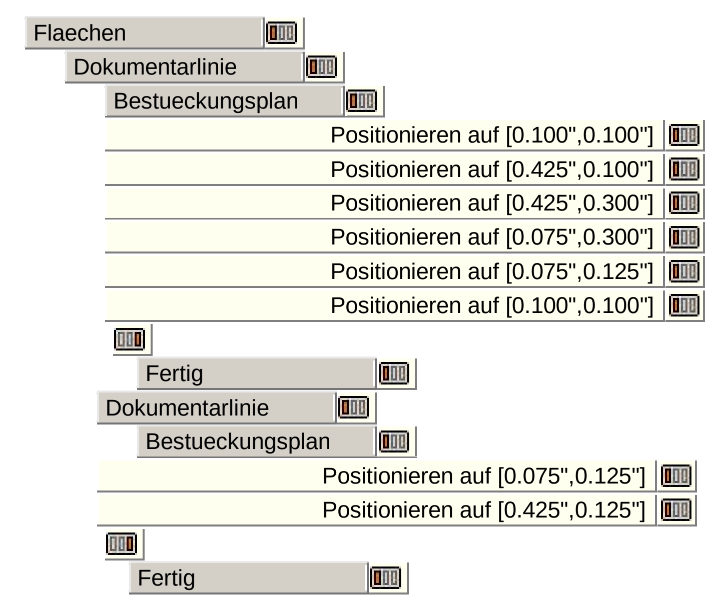
Definieren Sie die Anschlüsse **1** bis **7** durch Laden und Platzieren des in Kapitel 4.2.2 erstellten Padstacksymbols **so**:



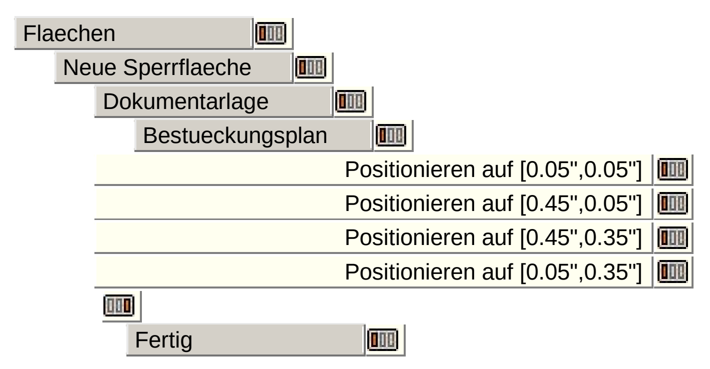
Definieren Sie die Anschlüsse **8** bis **14** durch Laden und Platzieren des Padstacksymbols **so**:



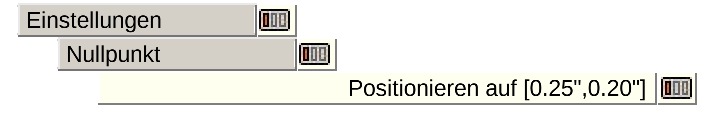
Definieren Sie mit den folgenden Komandos zwei Dokumentarlinien auf der Dokumentarlage Bestueckungsplan (Seite 2) zur Kennzeichnung des Bauteilumrisses:



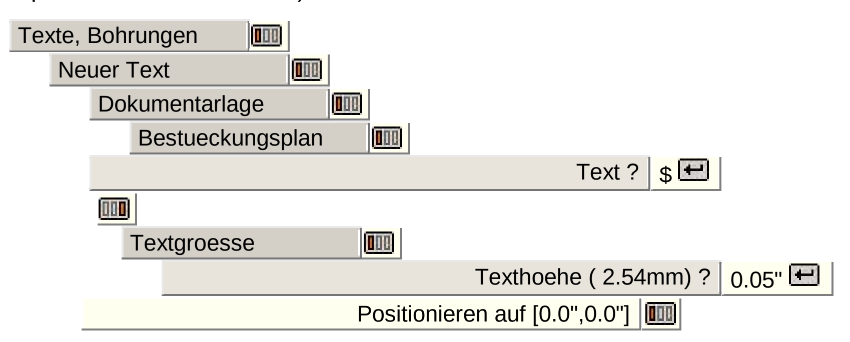
Die Einhaltung bestimmter Abstände zwischen SMD-Bauteilen ist häufig eine Forderung der Fertigung, durch die verhindert werden soll, dass durch die Lötwelle im Schwallbad sogenannte Lötbrücken (und damit Kurzschlüsse) generiert werden. Die Definition entsprechender "Bauteil-Sperrflächen" auf einer Dokumentarlage bewirkt, dass der Online-Check des **Layouteditors** Abstandsfehler meldet, wenn sich derartige Sperrflächen überlappen, d.h. die geforderten Bauteilabstände unterschritten werden. Definieren Sie mit den folgenden Kommandos eine Sperrfläche für die Bauteil-Abstandsprüfung auf der Dokumentarlage Bestueckungsplan (Seite 2):



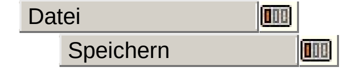
Setzen Sie mit dem folgenden Kommando den Bauteilnullpunkt auf den Mittelpunkt des Symbols:



Tragen Sie mit den folgenden Kommndos auf der Dokumentarlage Bestueckungsplan (Seite 2) den Text **$** (positioniert auf Bauteilnullpunkt mit Texthöhe 0.05") ein:

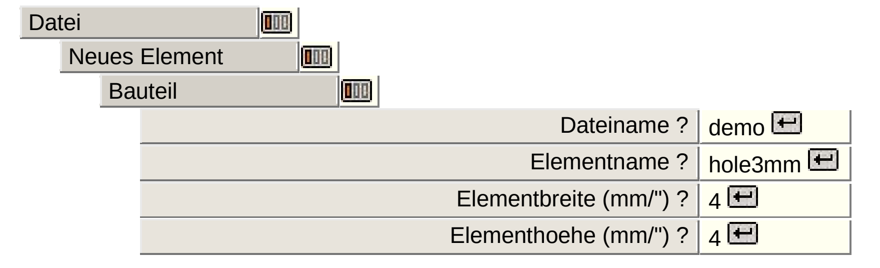


Vergessen Sie nicht, das soeben erstellte *Symbol* zu *speichern*:

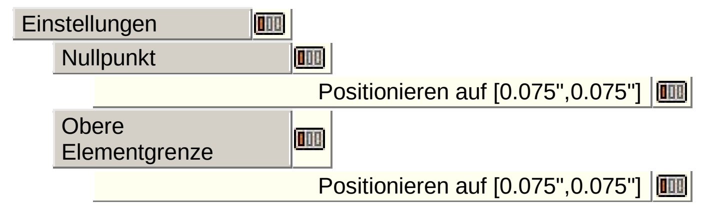


Definition eines konstruktiven Bauteilsymbols

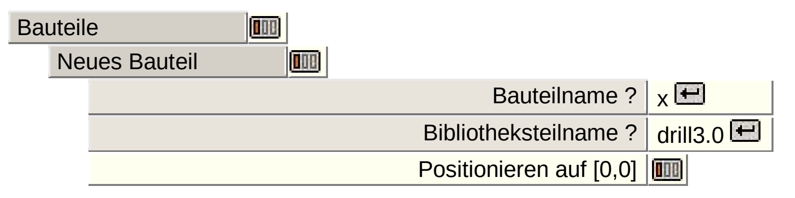
Generieren Sie mit den folgenden Kommandos in der Datei **demo.ddb** ein Bauteil mit dem Elementnamen **hole3mm** und einer Elementgröße von 4*4mm:



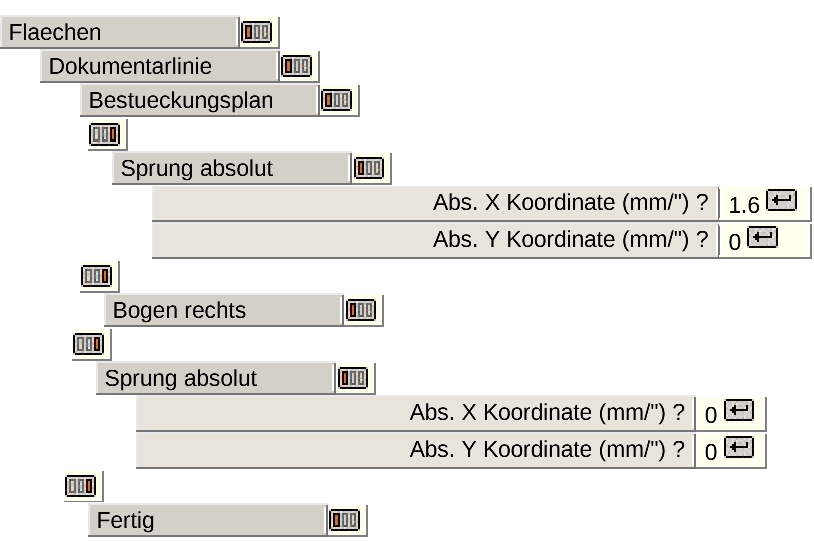
Setzen Sie den Nullpunkt und verändern Sie die obere Elementgrenze mit den folgenden Kommandos:



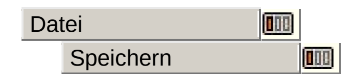
Definieren Sie den Anschluss **x** durch Laden und Platzieren des in Kapitel 4.2.2 erstellten Padstacksymbols **drill3.0**:



Definieren Sie mit den folgenden Kommandos eine kreisförmige Dokumentarlinie auf der Dokumentarlage Bestueckungsplan (Seite 2) zur Kennzeichnung des Bauteilumrisses:



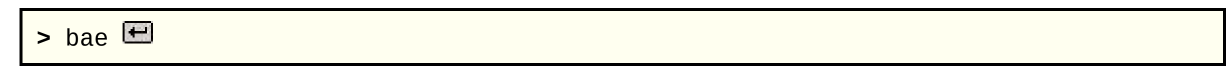
Vergessen Sie nicht, das soeben erstellte *Symbol* zu *sichern*:



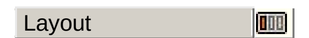
4.3 Layouterstellung

In diesem Abschnitt wird anhand des in den vorhergehenden Kapiteln bearbeiteten Beispiels die Layouterstellung mit den Layoutmodulen des **Bartels AutoEngineer** beschrieben. Dabei wird in der DDB-Datei **`demo.ddb`** ein Layout mit den Elementnamen **`board`** erstellt.

Wechseln Sie zunächst in das Verzeichnis, in dem die DDB-Datei **`demo.ddb`** abgelegt ist und starten Sie den **AutoEngineer**:



Wählen Sie den Menüpunkt Layout mit der Maus an, und bestätigen Sie Ihre Wahl durch Drücken der linken Maustaste:

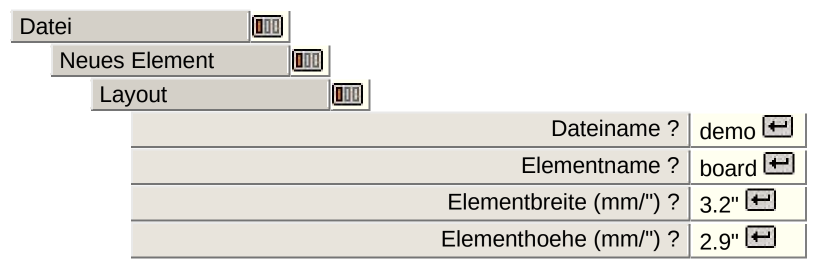


Nun wird der **Layouteditor** des **AutoEngineer** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

4.3.1 Erstellen und Bearbeiten von Layouts

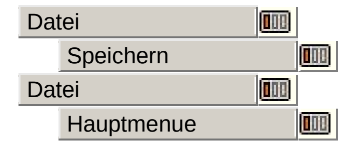
Erstellen eines neuen Layouts

Erzeugen Sie im DDB-File **`demo.ddb`** mit den folgenden Kommandos ein neues Layout mit dem Elementnamen **`board`** und einer Elementgröße von 3.2*2.9 Zoll:



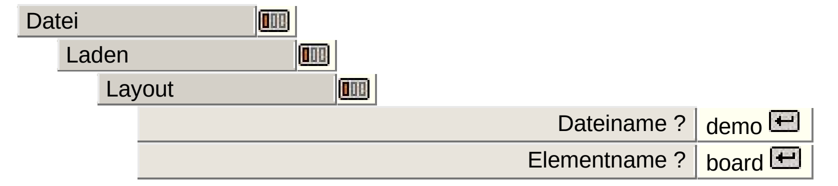
Nun sollte als heller Rahmen ein neues, allerdings noch leeres Layout auf dem Bildschirm erscheinen. Gibt das System z.B. die Fehlermeldung **`Dieses Element existiert bereits!`** aus, dann existiert das Layout **`board`** schon in der Datei **`demo.ddb`**. In diesem Fall ist es nicht möglich, dieses Layout neu zu generieren, sondern es muss geladen werden (siehe unten). Auf die Abfrage nach dem Elementnamen des Layouts sollte unbedingt der Name der Netzliste angegeben werden, für die das Layout zu erstellen ist, denn nur dann ist es dem System möglich, die Layoutdaten mit den Netzlistendefinitionen zu korrelieren.

Sie haben nun bereits ein neues Layout erzeugt. Speichern Sie dieses ab, und springen Sie zurück in die BAE-Shell:



Bearbeiten eines bestehenden Layouts

Rufen Sie nun erneut das Layoutsystem auf, und laden Sie mit folgenden Kommandos das im vorhergehenden Arbeitschritt erzeugte Layout; da das Layoutelement bereits existiert, können sowohl der Dateiname als auch der Elementname wahlweise auch über das Popupmenü durch Mausklick selektiert werden:



Nun erscheint das bereits erstellte Layout auf dem Schirm. Beim Versuch, ein Element zu laden, meldet das System **`Datei nicht gefunden!`**, wenn die DDB-Datei nicht existiert, bzw. **`Plan nicht gefunden!`**, wenn das gewählte Element nicht in der DDB-Datei gefunden werden konnte.

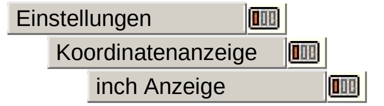
Beim Aufruf des **Layouteditors** ist dem System der Dateiname des zuvor in einem anderen BAE-Programm-Modul bearbeiteten Elements bekannt. Die Spezifikation dieses systemweiten Projektnamens kann durch Selektion des Buttons Projekt im Dateinamens-Popupmenü oder durch die Eingabe eines Leerstrings (Betätigen der Eingabetaste) auf die Abfrage nach dem Dateinamen erfolgen.

Die Selektion des Buttons Projekt im Elementnamens-Popupmenü oder die Eingabe eines Leerstrings (Betätigen der Eingabetaste) auf die Abfrage nach dem Elementnamen bewirkt, dass das System automatisch den über das Setup voreingestellten Defaultnamen für das Layoutelement verwendet (siehe hierzu auch die Beschreibung des Utilityprogramms **BSETUP** im Kapitel 7.2 dieses Handbuchs).

Das Layout wird beim Laden dynamisch aufgebaut. So wird zunächst das Layout selbst in den Speicher gelesen und dann (sofern bereits auf dem Layout vorhanden) die zugehörigen Bauteilelemente aus derselben Projektdatei. Diese wiederum laden die benötigten Padstacks. Die Padstacks ihrerseits laden die darauf definierten Pads. Existiert in der Projektdatei eine Netzliste, deren Name *identisch* mit dem Layoutelementnamen ist, dann wird auch diese geladen (Connectivity Generierung). Bei vorhandener Netzliste sind die Prüfroutinen des Systems in der Lage, das Leiterplatten-Design in jeder Phase des Entwurfs mit den Vorgaben in der Netzliste zu korrelieren. So hält das System ständig die Informationen über fehlende Bauteile, offene Verbindungen, Kurzschlüsse, usw. für den Anwender bereit.

Gibt das System beim Laden des Layouts die Meldung `Einige angeschlossene Pins fehlen!` aus, dann bedeutet dies, dass noch nicht alle in der Netzliste definierten Bauteile auf dem Layout platziert sind.

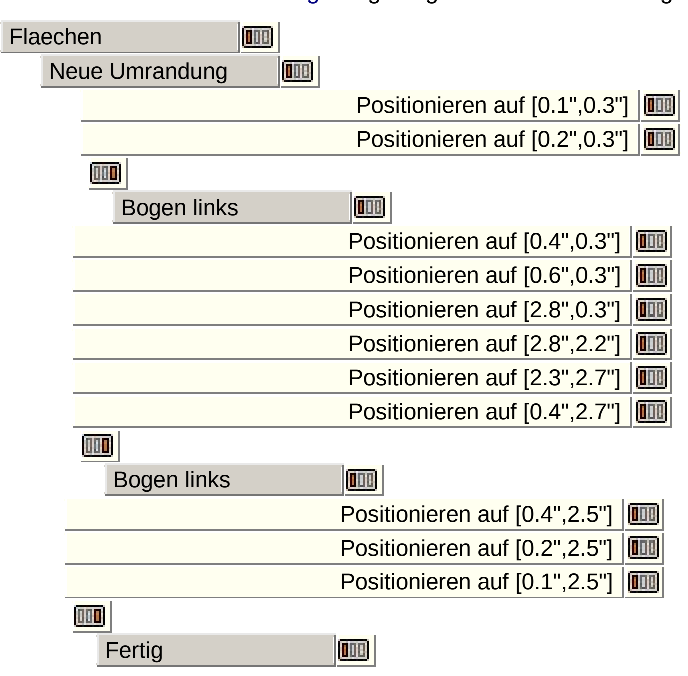
Stellen Sie vor den nächsten Arbeitschritten die Koordinatenanzeige auf Inch ein:



Die Koordinaten für die Positionierung des Kursors lassen sich nun rechts oben im Info-Feld in der Maßeinheit Inch ablesen.

Platinenumrandung

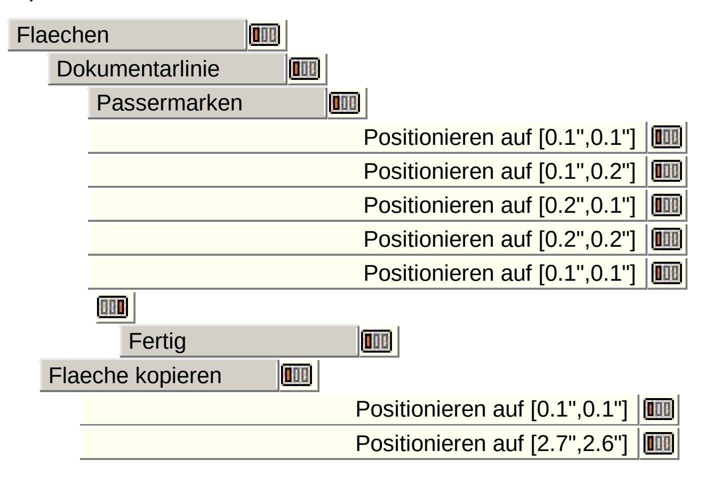
Bevor Sie mit der Platzierung der Bauteile beginnen, sollten Sie die Platinenumrandung festlegen. Definieren Sie mit den folgenden Kommandos die in Abbildung 4-2 gezeigte Platinenumrandung:



Die Platinenumrandung ist nicht zu verwechseln mit den Elementgrenzen. Während die Elementgrenzen die Aufgabe haben, die Größe des (Layout-)Elements definiert zu halten, stellt die Platinenumrandung sowohl die Begrenzung für die Platzierung als auch für das Verlegen von Leiterbahnen dar. Die Definition der Platinenumrandung ist für z.B. für den **Autorouter**-Lauf zwingend notwendig. Darüber hinaus schlägt der **Autorouter**-Aufruf auch fehl, wenn Bauteile außerhalb der Platinenumrandung platziert wurden.

Passermarken

Definieren Sie nun mit den folgenden Kommandos auf der Dokumentarlage Passermarken (wie in Abbildung 4-2 dargestellt) zwei Passermarken:



Die Passermarken sind Kontrollsymbole, die der Justierung der später ausgegebenen Plots dienen. Die Dokumentarlage für die Passermarken kann mit Hilfe des Setups festgelegt werden (siehe Programm **BSETUP**, Kapitel 7.2).

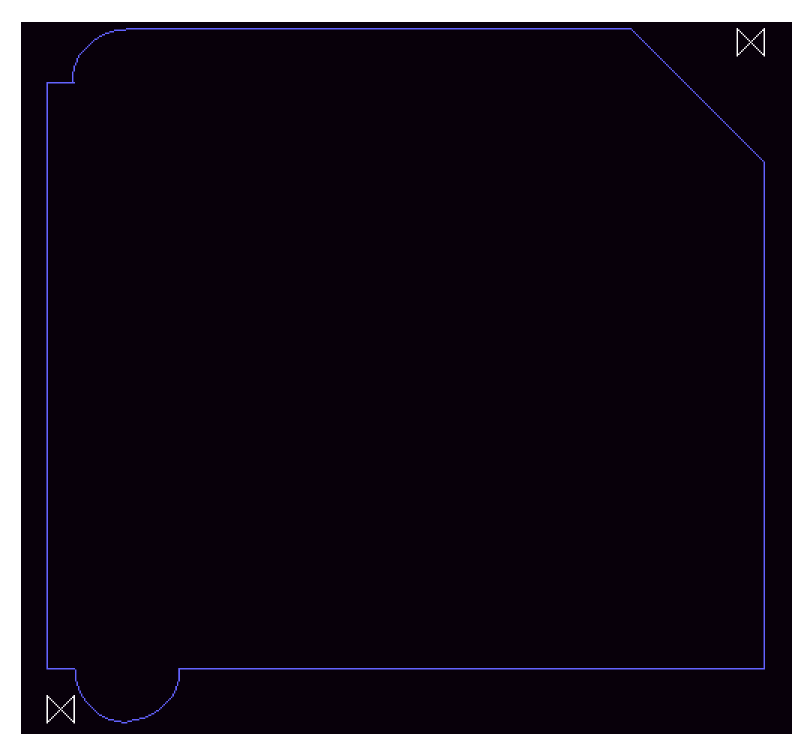
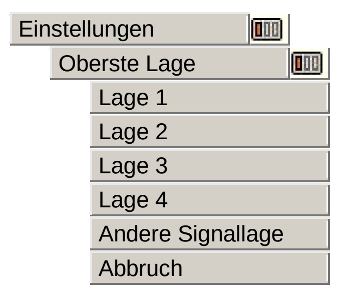


Abbildung 4-2: Layout mit Platinenumrandung und Passermarken

Oberste Lage

Im Layout kann eine Signallage zur Obersten Lage definiert werden. Diese Lage ist sehr nützlich, wenn die Lagenzahl oder die Lagenzuordnung bei der Erstellung der Bibliothek noch nicht feststeht. Die Zuweisung wird im Menü Einstellungen über die Funktion Oberste Lage durchgeführt und mit dem Layout gespeichert:



Nach der Erzeugung eines neuen Layouts ist die Oberste Lage per Default auf die Signallage 2 eingestellt. Die Oberste Lage definiert auch die Position des Spiegels beim Spiegeln z.B. eines SMD-Bauteils. Es wird stets gegen die so festgelegte Lage gespiegelt, alle Lagen oberhalb der Obersten Lage bleiben unberührt. Im **Autorouter** wird die Routinglagenzahl automatisch mit der Obersten Lage gleichgesetzt.

Grundsätzlich bestehen zwei Philosophien für die Definition der Obersten Lage. Zum einen ist es möglich, jeweils bei der Layouterstellung die Oberste Lage explizit zu definieren. Dies setzt allerdings den konsistenten Aufbau der Bibliothek voraus. D.h. in der Bibliothek müssen alle auf der Bauteilseite zu definierenden Elemente (wie z.B. SMD-Anschlüsse) auch tatsächlich der Obersten Lage zugewiesen werden, und darüber hinaus müssen die Definitionen für die Innenlagen bei zu erstellenden Multilayer-Layouts richtig durchgeführt werden. Zur Vereinfachung der Pflege und Definition von Bibliothekselementen stellt Ihnen das System die beiden Speziallagen Alle Lagen und Innenlagen zur Verfügung. Die Signallage Alle Lagen dient der Definition von Objekten, die auf allen Signallagen (von Signallage 1 bis einschließlich Oberste Lage) erscheinen sollen. Die Signallage Innenlagen dient der Definition von Padformen, die auf allen Signal-Innenlagen (zwischen der Signallage 1 und der Obersten Lage) erscheinen sollen. Daneben besteht die Möglichkeit, die Oberste Lage stets mit der Signallage 2 gleichzusetzen und die Lagen 3, 4, usw. als die Innenlagen zu betrachten. Dies erleichtert möglicherweise die Pflege der Layoutbibliothek, der Anwender muss allerdings bei Layouts mit mehr als 2 Signallagen vor dem **Autorouter**-Start die Routinglagenzahl explizit einstellen.

Mindestabstände

Die Mindestabstände für die Abstandsüberprüfungen durch den Design Rule Check werden im Menü Einstellungen mit den Abstandsfunktionen festgelegt und mit dem Layout abgespeichert:



Mit der Funktion Abstand LB/LB wird der Mindestabstand von Leiterbahn zu Leiterbahn festgelegt. Mit der Funktion Abstand LB/CU lässt sich der Mindestabstand von Leiterbahn zu Kupfer (z.B. Lötaugen) definieren. Die Funktion Abstand CU/CU legt den Mindestabstand von Kupfer zu Kupfer (z.B. Kupferfläche zu Lötauge) fest. Bei der Erzeugung eines neuen Layouts sind alle diese Mindestabstandswerte per Default auf 0.3mm gesetzt. Die Werte für die Mindestabstände wirken im Design Rule Check und sollten daher vor dem Platzieren von Bauteilen, dem Verlegen von Leiterbahnen, usw. entsprechend den Fertigungsvorgaben definiert werden. Beachten Sie, dass eine falsche (weil z.B. zu tolerante) Einstellung der Abstandsparameter möglicherweise zu fehlerhaften Layouts und damit zu einem mit erheblichen Kosten verbundenen Nachbearbeitungsaufwand führen kann.

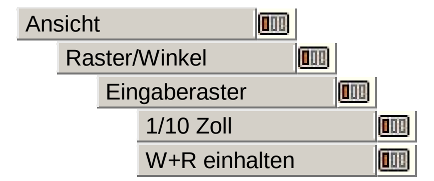
Netzbezogene Mindestabstände, die durch das Netzattribut `MINDIST` ggf. vorgegeben wurden, werden vom Design Rule Check bei der Prüfung der Abstandshaltung zu Leiterbahnen bzw. Potentialflächen gesondert berücksichtigt.

4.3.2 Bauteile, Platzierung

Das Menü Bauteile bietet die Möglichkeit, Gehäusebauformen aus selektierbaren Layoutbibliotheken auf das Layout zu laden und diese zu platzieren. Auch können einmal platzierte Bauteile wieder gelöscht werden.

Eingaberaster

Grundsätzlich können im **Bartels AutoEngineer** alle Grafikeingaben in beliebigen Rastern oder auch rasterfrei erfolgen. Dennoch sollte die Bauteilplatzierung in einem definierten Raster vorgenommen werden, um sicherzustellen, dass das spätere Verlegen der Leiterbahnen in einem geeigneten Raster erfolgen kann. Um hohe Packungsdichten auf der Leiterkarte zu erreichen, sollte es z.B. im später festzulegenden Routingraster für den **Autorouter** möglich sein, Leiterbahnen zwischen benachbarten Bauteilanschlüssen zu verlegen. Stellen Sie also zunächst mit folgenden Kommandos das Eingaberaster auf 1/10 Zoll ein:



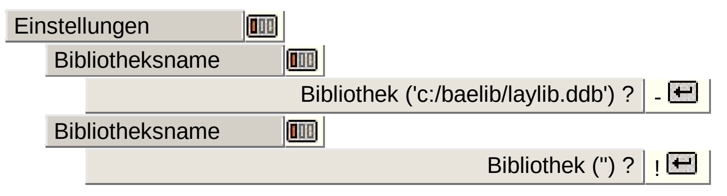
Bibliothekszugriff

Die Funktion Bibliotheksname aus dem Menü Einstellungen ermöglicht die Selektion der Layoutbibliothek, aus der die Gehäusebauformen geladen werden sollen. Überprüfen Sie zunächst mit folgenden Kommandos, wie der Bibliothekspfad gesetzt ist:



Im Prompt zur Abfrage nach der Bibliothek zeigt das System den Namen der aktuell eingestellten Bibliothek an. Nach dem Aufruf des **Layouteditors** ist dies zunächst der Name der über das Setup eingestellten Standard-Layoutbibliothek (siehe hierzu die Beschreibung des Utilityprogramms **BSETUP** im Kapitel 7.2 dieses Handbuchs).

Durch die Eingabe eines leeren Strings auf die Abfrage nach der Bibliothek ändert sich der Bibliotheksname nicht. Durch die Eingabe von **-** wird der Bibliotheksname zurückgesetzt, d.h. es ist dann keine Bibliothek selektiert. Die Eingabe von **!** bzw. **.** setzt den Bibliotheksnamen wieder auf die durch das Setup eingestellte Bibliothek. Überprüfen Sie dies mit folgenden Kommandos:



In den Windowsversionen der BAE-Software erfolgt die Abfrage des Bibliotheksnamens über einen Dateiabfragedialog.

Das Datenbankkonzept des **Bartels AutoEngineer** impliziert, dass auch jede Projektdatei als Bibliothek fungieren und als solche im System angemeldet werden kann. Aktiviert der Anwender die Funktion zum Laden eines Bauteils, dann sucht das System zunächst innerhalb der aktuellen Projektdatei nach dem angeforderten Bibliotheksteil. Ist das Element nicht hierin schon enthalten, wird die Suche in der im System angemeldeten Default-Bibliothek fortgesetzt. Immer, wenn ein Bauteil aus einer Bibliothek auf das Layout geladen wird, erstellt das System automatisch eine Kopie des entsprechenden Layoutsymbols in der aktuellen Projektdatei. Das Layoutsymbol ist dann in der aktuellen Projektdatei abgespeichert, wird also bei mehrfachem Platzieren nicht mehr aus der entsprechenden Bibliothek geholt. Abbildung 4-3 verdeutlicht das Schema des Bibliothekszugriffs innerhalb des Layoutsystems.

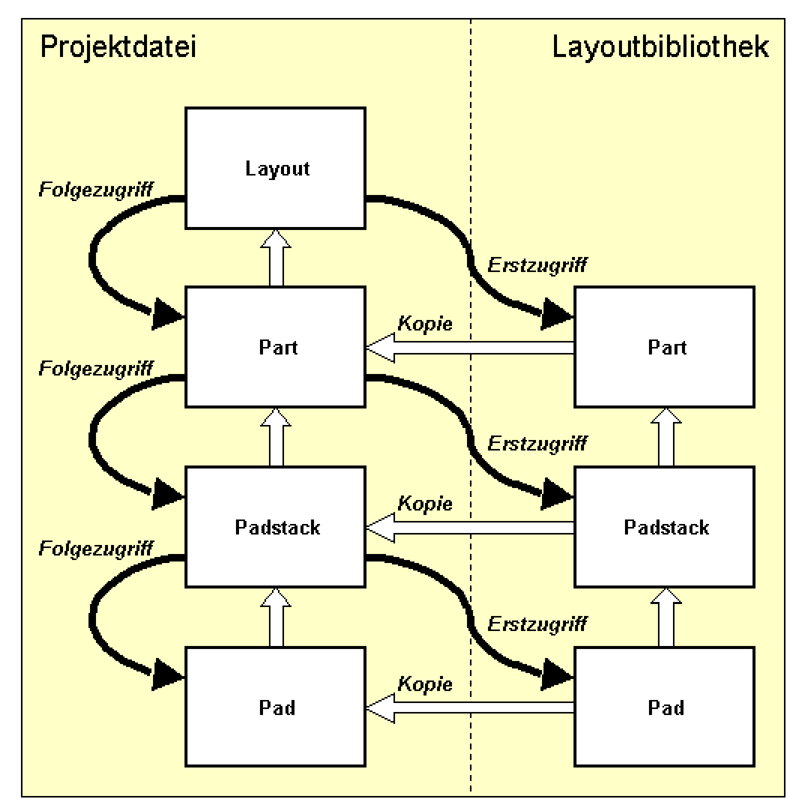
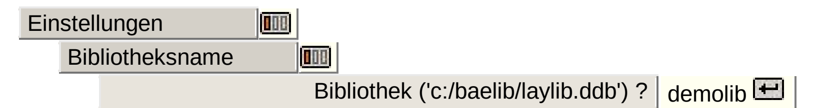


Abbildung 4-3: Layout-Bibliothekszugriff

Das Umschalten der Default-Bibliothek innerhalb des Systems ist im Grunde nur dann notwendig bzw. sinnvoll, wenn Bauteilsymbole aus einer nicht über den Bibliothekspfad erreichbaren Bibliothek zum ersten Mal im Layout platziert werden sollen, wie in unserem Fall z.B. aus der Bibliotheksdatei **`demolib.ddb`** im aktuellen Verzeichnis. Stellen Sie nun den Bibliothekspfad auf **`demolib.ddb`** ein:



Durch obige Eingabe ist die DDB-Datei **`demolib.ddb`** als aktuelle Layoutbibliothek im System angemeldet, und es können nun Layoutsymbole aus dieser Bibliothek auf das aktuell geladene Layout geladen werden.

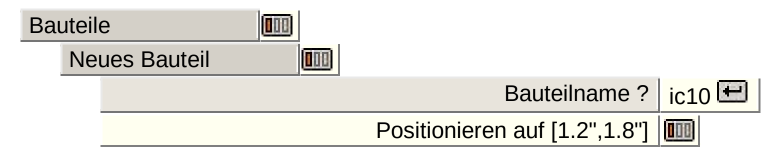
Manuelle Bauteilplatzierung

Zur manuellen Bauteilplatzierung stehen die Funktionen Neues Bauteil, Folgebauteil und Bauteile platzieren aus dem Menü Bauteile zur Verfügung. Diese Funktionen berücksichtigen die aktuell zur Platzierung selektierte Bauteilmenge. Nach dem Laden eines Layouts sind zunächst alle Bauteile selektiert. Mit den Funktionen des Untermenüs Bauteilmenge lässt sich die Menge der Bauteile für die Platzierung beschränken. Sind bei Aufruf der Funktion Folgebauteil bereits alle selektierten Bauteile platziert, dann wird die Fehlermeldung **Alle selektierten Bauteile sind bereits platziert!** angezeigt. Diese Meldung weist gleichzeitig darauf hin, dass noch unselektierte, unplatzierte Bauteile vorhanden sind. Im Popupfenster für die Bauteilnamensauswahl sind platzierte selektierte Bauteile als **[name]** aufgeführt, unplatzierte selektierte Bauteile als **: name**, unplatzierte unselektierte Bauteile als **< name >** und platzierte unselektierte Bauteile als **(name)**. Weitere Informationen zur Definition von Bauteilmengen für die Platzierung finden Sie in Kapitel 4.4.1.

Mit der Funktion Neues Bauteil aus dem Menü Bauteile können Bauteile auf das aktuell bearbeitete Layout geladen und platziert werden. Auf Layoutebene wird mit der Abfrage nach dem Bauteilnamen ein Popupmenü mit der Liste der Bauteilnamen aus der Netzliste aktiviert. Hierin werden bereits platzierte Bauteile in eckigen Klammern (**[**, **]**) und unplatzierte Bauteile nach einem Doppelpunkt (**:**) angezeigt. Wird der Name eines unplatzierten Netzlistenbauteils selektiert bzw. spezifiziert, dann wird dieses Bauteil mit der zugehörigen Gehäusebauform geladen und kann platziert werden. Bei Selektion bzw. Spezifikation eines bereits platzierten Bauteils wird eine Neuplatzierung nur dann durchgeführt, wenn der Anwender dies in einer entsprechenden Abfrage ausdrücklich bestätigt. Die Eingabe eines Leerstrings bzw. die Selektion des Buttons Unplaz. im Popupmenü bewirkt die Selektion des nächsten noch nicht platzierten Bauteils aus der Netzliste. Dieser Automatismus wird auch mit der Funktion Naechstes Bauteil angeboten und funktioniert solange, bis alle Netzlistenbauteile (aus der Bauteilmenge) platziert sind. Bei Angabe eines nicht in der Netzliste enthaltenen Bauteilnamens (d.h. bei Spezifikation eines Konstruktivbauteils durch die Eingabe eines nicht im Popupmenü angezeigten Namens) wird zusätzlich eine Abfrage nach dem gewünschten Bibliothekselement aktiviert. Hierbei wird zunächst ein Popupmenü zur Auswahl der Bibliotheksdatei angeboten, wobei die Dateinamensliste aus dem im BAE-Setup definierten Pfadnamen für die Layoutbibliothek abgeleitet wird, d.h. es werden alle im Verzeichnis der im System angemeldeten Standardlayoutbibliothek enthaltenen DDB-Dateien aufgelistet. Bei der Auswahl der Layoutbibliothek kann über den Button Bibl. bzw. durch Eingabe von **>** die aktuell über die Funktion Bibliotheksname aus dem Menü Einstellungen selektierte Standardlayoutbibliothek gewählt werden. Mit Projekt wird die aktuell bearbeitete DDB-Datei selektiert. Nach Auswahl der Layoutbibliothek erfolgt die Abfrage nach dem gewünschten Bibliothekselementnamen (mit entsprechendem Popupmenüü). Bibliothekselemente können wahlweise auch direkt durch Eingabe des Bibliotheksdateinamens, eines Schrägstrichs (**/**) und des Elementnamens spezifiziert werden (ein Fragezeichen anstelle des Elementnamens aktiviert hierbei wiederum ein Popupmenü mit den in der angegebenen Bibliotheksdatei enthaltenen Bibliothekslementen). Durch die Eingabe eines Leerstrings auf die Abfrage nach dem Bibliothekselementnamen wird das zuletzt mit Neues Bauteil selektierte Bibliothekselement ausgewählt (sofern zuvor bereits ein Bibliothekselement spezifiziert wurde).

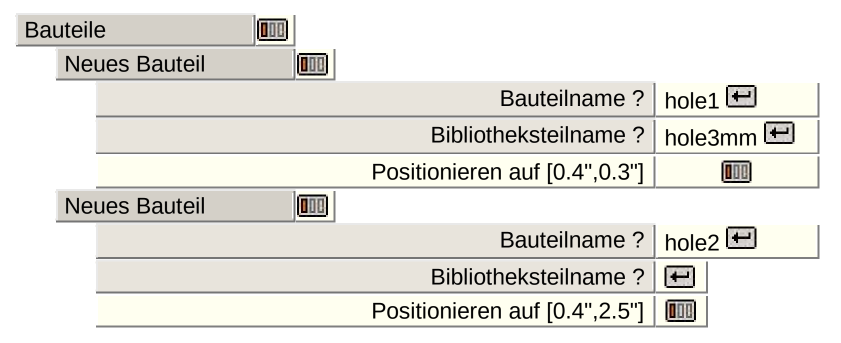
Beachten Sie, dass die Popupmenüs zur Auswahl des Bauteilnamens auch in den entsprechenden Routinen der Funktionen Name aendern und Name in Netzliste integriert sind, und dass die Popupmenüs zur Auswahl des Bibliothekselements in der Funktion Neues Bauteil auch für die Platzierung von Pins (d.h. zur Selektion von Padstacks) auf Bauteilebene, sowie zum Laden von Pads auf Padstackebene angeboten werden.

Laden und platzieren Sie mit den folgenden Kommandos das Bauteil **IC10**:



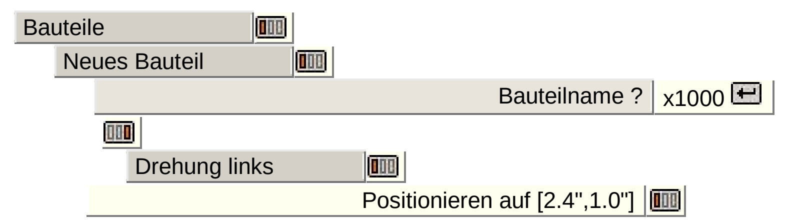
Nach Angabe des Bauteilnamens überprüft das System, ob ein Bauteil mit diesem Namen in der Netzliste existiert. Ist dies der Fall, dann wird das in der Netzliste zugewiesene Gehäuse (in obigem Beispiel **dil14**) aus der Bibliothek geladen und kann platziert werden. Sollte das System hierbei die Meldung **Plan nicht gefunden!** ausgeben, dann ist das benötigte Layoutsymbol weder in der Projektdatei noch in der eingestellten Bibliothek verfügbar. In diesem Fall sollten Sie überprüfen, ob Sie auf die richtige Layoutbibliothek zugreifen (Funktion Bibliotheksname im Menü Einstellungen).

Laden und platzieren Sie mit den folgenden Kommandos die beiden Bauteile **`hole1`** und **`hole2`** mit der in Kapitel 4.2.3 in der Projektdatei **`demo.ddb`** erstellten Gehäusebauform **`hole3mm`**:



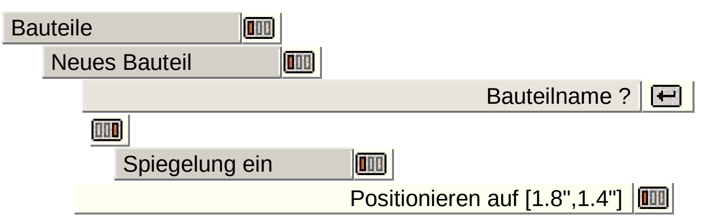
Die beiden (konstruktiven) Bauteile **`hole1`** und **`hole2`** sind nicht in der Netzliste definiert, d.h. auch die dafür zu verwendende Gehäusebauform ist dem System nicht bekannt. Daher erfolgt für diese Bauteile jeweils eine Abfrage nach dem Bibliotheksteilnamen. Hierbei wird zunächst ein Popupmenü mit allen im Bibliotheksverzeichnis verfügbaren Bibliotheksdateien angezeigt. Mit dem Button Projekt ist dabei der Zugriff auf die Layoutsymbole der aktuellen DDB-Datei, d.h. auf die Projektdatenbank möglich. Nach der Auswahl einer der angebotenen Bibliotheksdateien erfolgt die Aktivierung eines weiteren Popupmenüs mit der Liste der in der selektierten Bibliothek verfügbaren Symbole. Bei Eingabe eines Leerstrings (Betätigen der Eingabetaste ↵) auf die Abfrage nach dem Bibliotheksteilnamen wird das zuletzt geladene Layoutsymbol verwendet.

Laden und platzieren Sie den Steckverbinder **`x1000`** mit den folgenden Kommandos:



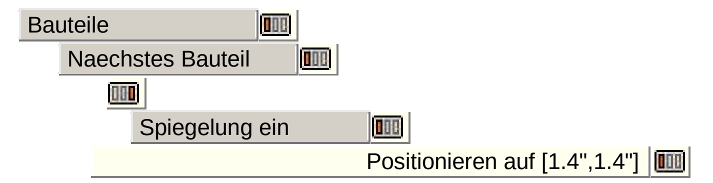
Das Untermenü, das während der Platzierung von Bauteilen über die rechte Maustaste aufgerufen werden kann, erlaubt die Positionierung des Bauteils auf eine Absolutkoordinate, die Drehung (auch um beliebige Winkel), oder die Spiegelung (auf die Lötseite; für SMD-Bauteile).

Platzieren Sie mit den folgenden Kommandos ein weiteres Bauteil:

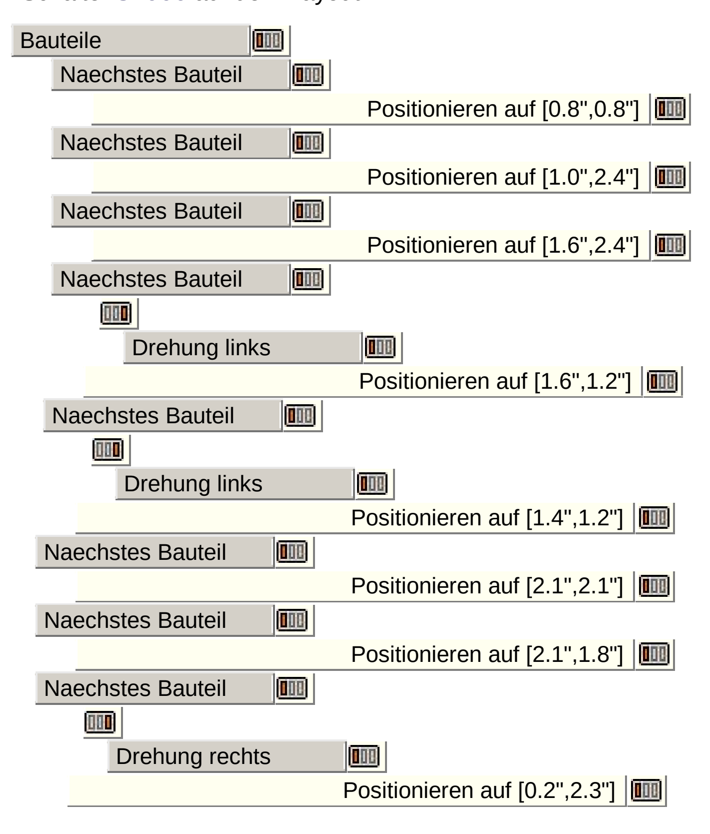


Ein Leerstring (Betätigen der Eingabetaste ↵) auf die Abfrage nach dem Bauteilnamen hat zur Folge, dass automatisch das nächste in der Netzliste definierte Bauteil geladen wird. In obigem Beispiel ist dies der Kondensator **`c100`**, der in SMD-Bauform ausgeführt ist und auf die Lötseite (gespiegelt) platziert wird.

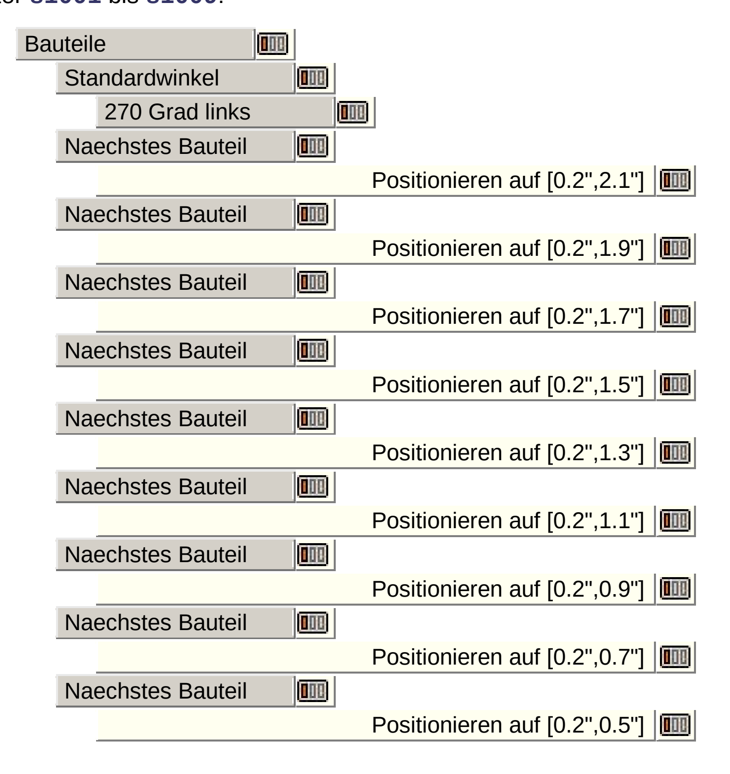
Der Automatismus zum Laden des nächsten in der Netzliste definierten Bauteils (ohne Kenntnis des Bauteilnamens) kann auch mit der Funktion Naechstes Bauteil aktiviert werden. Platzieren Sie mit Hilfe dieser Funktion das Bauteil **`c101`**:



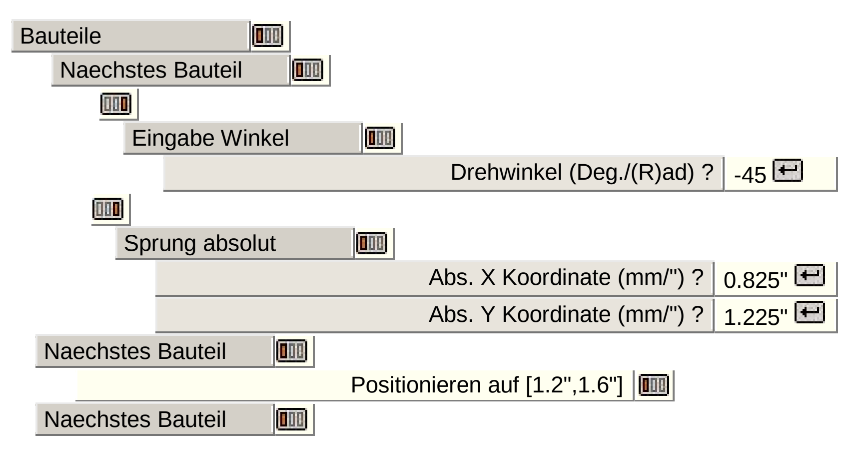
Platzieren Sie mit den folgenden Kommandos das Relais **`k1`**, die Widerstände **`r100`**, **`r101`**, **`r102`**, **`r103`**, **`r104`** und **`r105`**, sowie den Schalter **`s1000`** auf dem Layout:



Sind nacheinander mehrere Bauteil zu laden, die gespiegelt (z.B. SMD-Bauteile) bzw. gedreht zu platzieren sind, dann empfiehlt es sich, Standardwinkel und Spiegelungsmodus für die Platzierung vorher entsprechend einzustellen. Setzen Sie mit den folgenden Kommandos den Standardwinkel für die Platzierung auf 270 Grad, und laden Sie die noch fehlenden Schalter **`s1001`** bis **`s1009`**:



Platzieren Sie die nun noch fehlenden Bauteile (**v1** und **v1000**) mit den folgenden Kommandos:



Die nach dem letzten Aufruf der Funktion Naechstes Bauteil ausgegebene Meldung **Alle Bauteile sind bereits platziert!** weist darauf hin, dass alle in der Netzliste definierten Bauteile auf dem Layout platziert sind.

Wenn Sie alle Arbeitsschritte korrekt ausgeführt haben, dann sollte das Layout nun entsprechend Abbildung 4-4 aussehen.

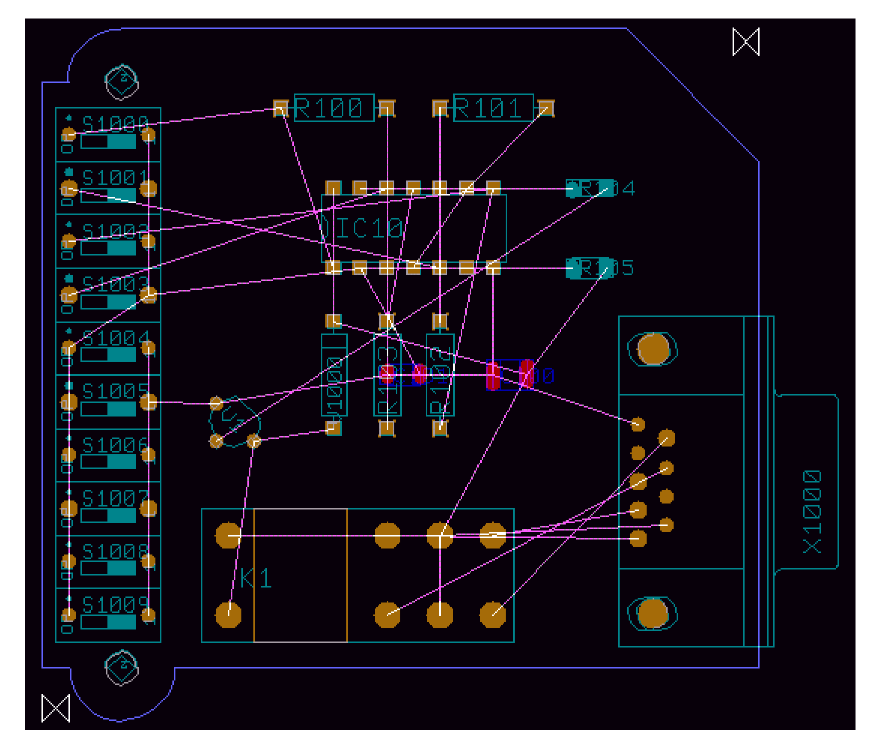


Abbildung 4-4: Layout mit platzierten Bauteilen

Bauteile bewegen und löschen

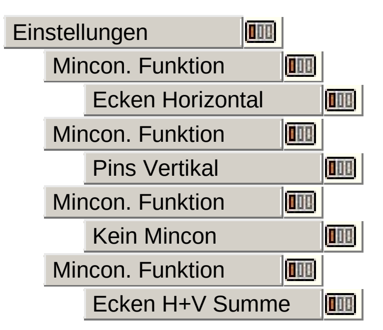
Mit Hilfe der Funktion Bewegen Bauteil können bereits platzierte Bauteile bewegt, gedreht und gespiegelt werden (die rechte Maustaste aktiviert ein entsprechendes Untermenü). Das Löschen eines platzierten Bauteils erfolgt mit der Funktion Loeschen Bauteil. Testen Sie diese Funktionen auf dem aktuell geladenen Layout. Bedienen Sie sich hierbei der Funktionen Undo und Redo, um wieder in den Zustand gemäß Abbildung 4-4 zurückzukehren.

Gehäusebauform ändern

Das während der interaktiven Bauteilplatzierung über die rechte Maustaste erreichbare Untermenü enthält die Funktion Bauform aendern. Diese Funktion ermöglicht die Änderung der Gehäusebauform des aktuell bearbeiteten Bauteils. Hierzu wird ein Auswahlmenü mit der Liste der über eine SCM-Symbol-Attributwertzuweisung an das Attribut `$plname` der Form `[bauform1,bauform2,...]` oder über **LOGLIB** (siehe Kapitel 7.11) definierten alternativen Gehäusebauformen angezeigt. Eine durch die Funktion Bauform aendern ausgegebene Fehlermeldung der Form `Es sind keine Alternativbauformen definiert!` besagt, dass für das aktuell bearbeitete Bauteil keine alternativen Gehäusebauformen definiert sind; die Fehlermeldung `Einige angeschlossene Pins fehlen!` wird ausgegeben, wenn die Pindefinitionen des selektierten Alternativgehäuses nicht mit den Netzlistenvorgaben für das bearbeitete Bauteil übereinstimmen. Nach Änderung von Bauformen wird beim Speichern des Layouts ein spezieller Datenbankeintrag erzeugt, der im **Schaltplaneditor** ausgewertet wird und die automatische Durchführung notwendiger Backannotation-Prozesse beim Laden von Schaltplänen aktiviert (siehe hierzu Kapitel 2.7. Im **Packager** wird dieser Datenbankeintrag ebenfalls ausgewertet, um den Anwender ggf. über eine Bestätigungsabfrage auf die Notwendigkeit zur Durchführung der Backannotation vor dem nächsten **Packager**-Lauf hinzuweisen (siehe hierzu Kapitel 3.2.3).

Mincon-Funktion, Airline-Anzeige

Zur Unterstützung des Anwenders beim manuellen Platzieren werden nicht realisierte Verbindungen als Luftlinien ("Airlines", "Unroutes") dargestellt. Die Mincon-Funktion sorgt dafür, dass diese Airlines dynamisch zum jeweils nächst gelegenen Anschlusspunkt aktualisiert werden. Achten Sie während des Bewegens von Bauteilen auf die Arbeitsweise der Mincon-Funktion. Überprüfen Sie mit den Kommandos



die verschiedenen Methoden zur Ermittlung des nächst gelegenen Anschlusspunktes, und ändern Sie über die Funktion Airlines Anzeige im Menü Bauteile dabei auch die Art der Luftliniendarstellung. Dabei wird mit der Option Keine Airlines die Airlineanzeige während der Bauteilbewegung deaktiviert. Mit Airlines Statisch werden die Luftlinien zwar (statisch) angezeigt, es unterbleibt jedoch eine dynamische Neuberechnung der jeweils kürzesten Unroutes entsprechend der aktuell eingestellten Mincon-Funktion wie sie mit der Option Airlines Dynamisch aktiviert werden kann. Die dynamische Airlineanzeige ist per Default eingestellt (beachten Sie hierbei, wie die Airlines während des Bewegens von Bauteilen zum jeweils nächstgelegenen Pin bzw. Kupfereckpunkt "springen").

Über die Mincon-Optionen Netz sichtbar und Netz unsichtbar kann eine netzspezifische Airlineanzeige aktiviert werden. Damit ist es möglich, spezielle Netze in der Airlineanzeige auszublenden (Netz unsichtbar) bzw. nach Bedarf wieder einzublenden (Netz sichtbar). Die Auswahl der ein- bzw. auszublendenden Netze erfolgt über ein Netznamenspopupmenü das wahlweise auch die Angabe von Netznamensmustern zur simultanen Selektion verschiedener Netze gestattet.

Die Mincon-Option Alle sichtbar erzwingt die grafische Anzeige aller Airlines, während mit Alle unsichtbar sämtliche Airlines ausgeblendet werden.

Während des Platzierens von Bauteilen stehen in dem über die rechte Maustaste erreichbaren Kontextmenu die Optionen Bauteilnetze und Alle Netze zur Auswahl zwischen bauteilspezifischer und globaler Airlineanzeige zur Vefügung.

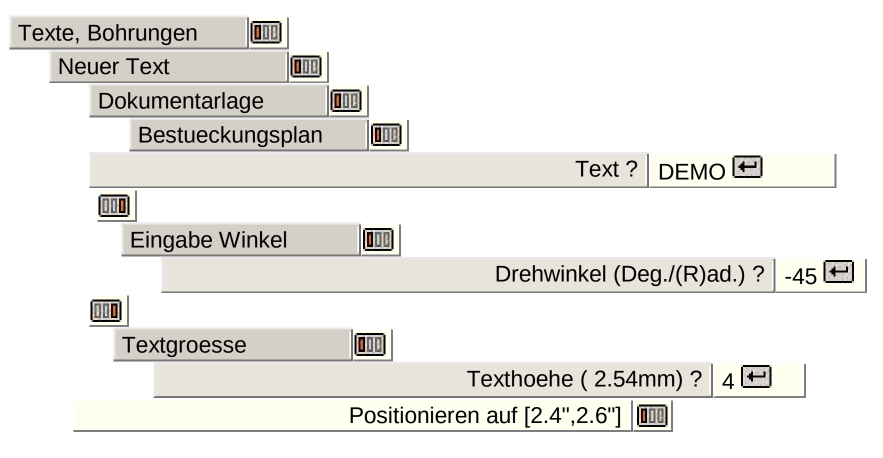
4.3.3 Text und Grafik

Auf Layoutebene können Texte (auf Dokumentar- oder Signallagen) und Grafiken (Dokumentarlinien, Dokumentarflächen, Kupferflächen, Potentialflächen, Sperrflächen, Füllbereiche) definiert werden.

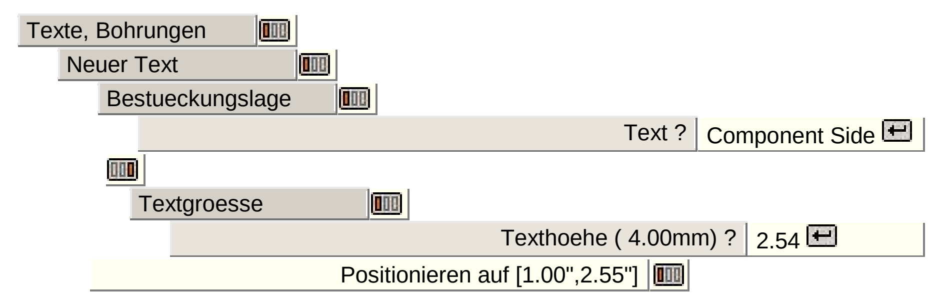
Stellen Sie vor den nächsten Arbeitsschritten (sofern nicht schon geschehen) die Koordinatenanzeige auf Inch (Menü Einstellungen) und das Eingaberaster auf 1/20 Zoll ein (Menü Ansicht).

Eingabe von Texten auf dem Layout

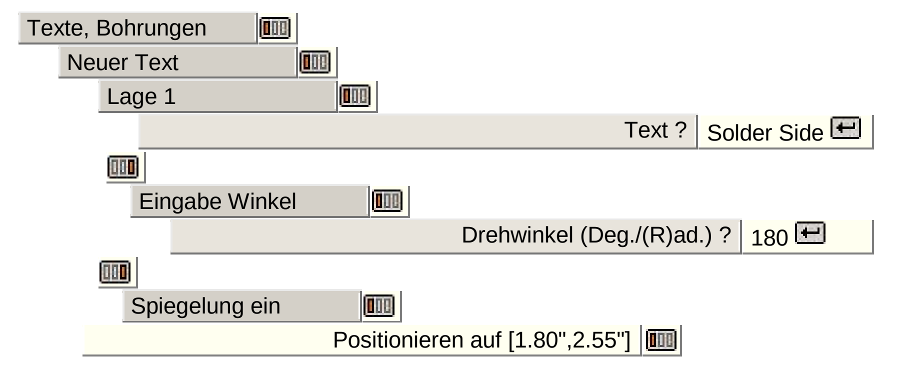
Definieren Sie mit den folgenden Kommandos den Text **`DEMO`** auf der Dokumentarlage Bestueckungsplan (Seite 2) mit einer Textgröße von 4mm, und platzieren Sie diesen Text rechts oben im Layout:



Definieren Sie mit den folgenden Kommandos auf der Bauteilseite (Oberste Lage bzw. Bestueckseite) den (Kupfer-)Text **`Component Side`**:



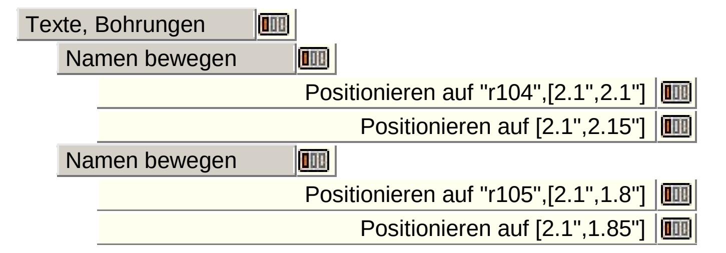
Die Einstellung der Textgröße bleibt als Default für nachfolgende Textdefinitionen erhalten. Definieren Sie mit den folgenden Kommandos, auf der Lötseite (Signallage 1) den (Kupfer-)Text **`Solder Side`** (gedreht und gespiegelt):



Auf Signallagen definierte Kupfertexte werden vom Design Rule Check geprüft, d.h. es werden Abstandsfehler gemeldet, wenn sich solche Texte mit anderem Kupfer (Lötaugen, Leiterbahnen, usw.) auf der derselben Signallage überschneiden.

Bewegen von Bauteilnamen und Bauteilattributtexten

Mit der Funktion Namen bewegen können Bauteilnamen verschoben werden. Bewegen Sie die Namen der beiden SMD-Widerstände `r104` und `r105` jeweils über das entsprechende Bauteil:

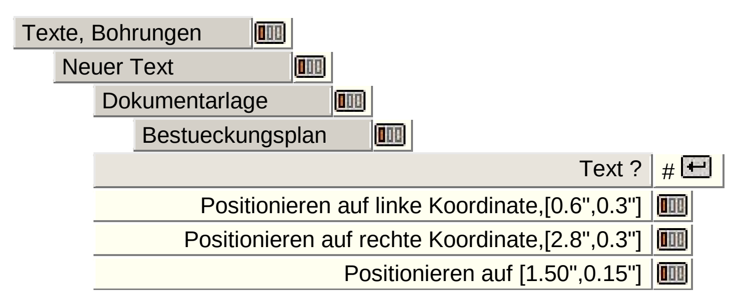


Mit der Funktion Attribut bewegen aus dem Menü Texte bzw. Bauteile können Bauteilattributtexte selektiv platziert bzw. bewegt werden. Die mit Attribut bewegen festgelegten Textoffsets haben Vorrang vor ggf. mit Name bewegen vorgegebenen globalen Textoffsets. Die Selektion des zu verschiebenden Attributs erfolgt durch Anklicken des Attributtexts. Der Bauteilname selbst gilt ebenfalls als Attribut (**$**) und kann somit mit der Funktion Attribut bewegen auch selektiv verschoben werden, ohne die Platzierung der übrigen Bauteilattribute zu beeinträchtigen.

Bauteiltexte gruppenselektierter Bauteile können mit der Funktion Bauteiltexte Reset des **User Language**-Programms **GEDGROUP** wieder auf die im Bauteilbibliothekselement definierten Defaultpositionen zurückgesetzt werden.

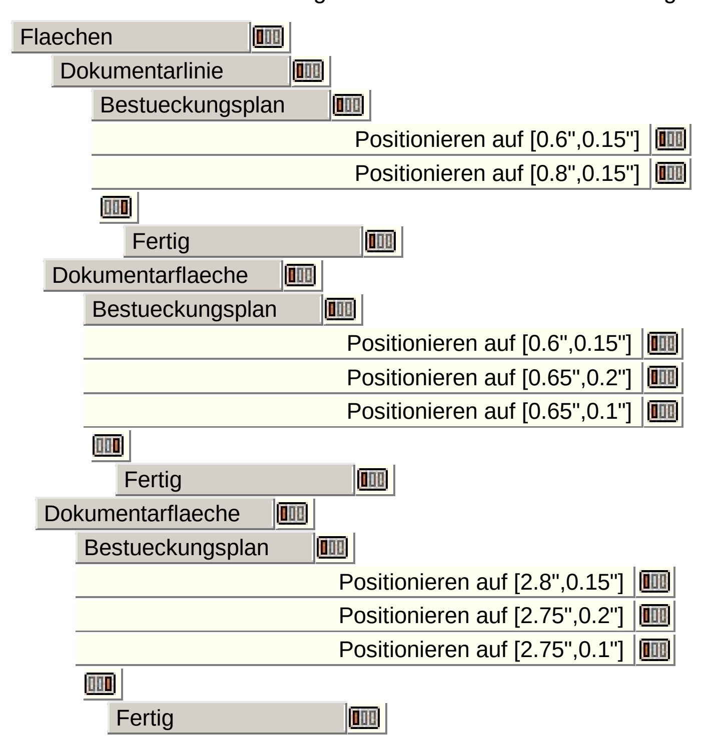
Bemaßung

Mit dem **#**-Zeichen als neuem Text wird eine Meßfunktion aktiviert, die die Distanz zweier danach abgefragter Punkte ermittelt und als Text darstellt. Bemaßen Sie mit den folgenden Kommandos auf dem Bestückungsplan die Länge der horizontalen Platinenkante unten rechts an der Leiterkarte (die Bemaßung wird in der Maßeinheit durchgeführt, die über die Koordinatenanzeige eingestellt ist, und sollte demnach also einen Wert von 2.2" erzeugen):



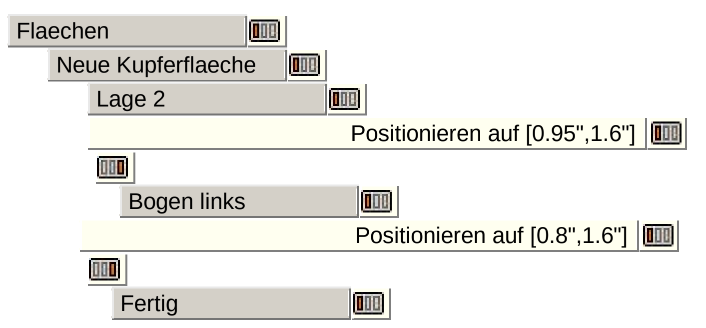
Dokumentarlinien, Dokumentarflächen

Tragen Sie auf dem Bestückungsplan mit den folgenden Kommandos eine Dokumentarlinie sowie zwei pfeilförmige Dokumentarflächen zur Kennzeichnung der zuvor definierten Bemaßung ein:



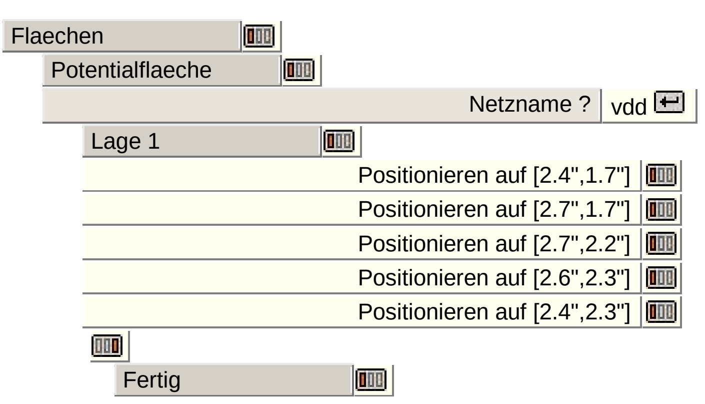
Kupferflächen

Definieren Sie mit den folgenden Kommandos eine kreisförmige Kupferfläche auf der Signallage 2:



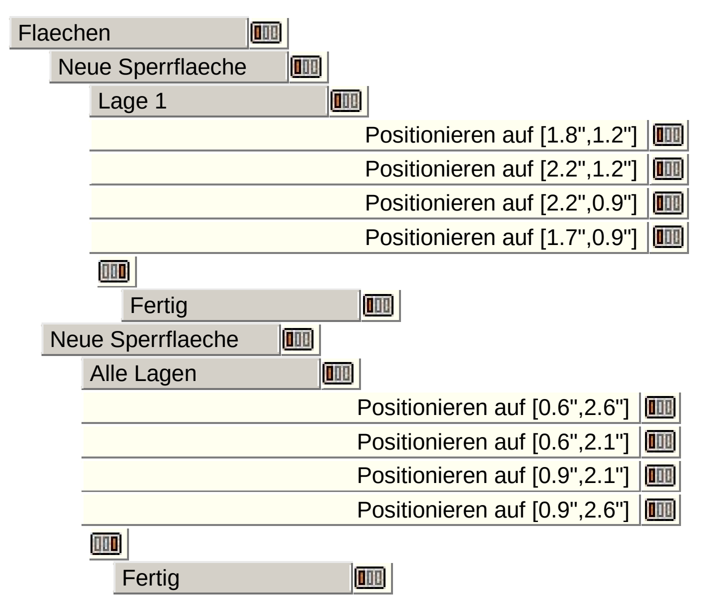
Potentialflächen

Definieren Sie mit den folgenden Kommandos auf der Signallage 1 eine Potentialfläche für das Signalnetz **Vdd** (d.h. eine Kupferfläche, die dem Netz **Vdd** zugeordnet ist, und die der **Autorouter** automatisch mit diesem Netz verbinden soll):



Sperrflächen

Sperrflächen legen Bereiche auf der Leiterkarte fest, auf welchen keine Leiterbahnen, Kupferflächen, usw. erlaubt sind. Definieren Sie mit den folgenden Kommandos eine Sperrfläche auf der Signallage 1 sowie eine Sperrfläche auf der Signallage Alle Lagen:



4.3.4 Leiterbahnen, Routing

Im Menü Leiterbahnen sind die Funktionen für das interaktive Routen, d.h. für das manuelle Verlegen von Leiterbahnen auf Layout- bzw. Bauteilebene enthalten. Es können neue Leiterbahnen erzeugt, bestehende geändert oder gelöscht werden. Es können die Leiterbreiten einzelner Leiterbahnsegmente, von Bahnen oder ganzen Netzen geändert werden. Auch können kritische Netze (z.B. Stromversorgungskamm) ganz oder teilweise vorverlegt und fixiert werden, um zu verhindern, dass der **Autorouter** die entsprechenden Leiterbahnen neu- oder umverlegt.

Grafik-Kontrollfunktionen, Eingaberaster, Bilddarstellung

Wird eine Funktion zum Verlegen oder Ändern einer Leiterbahn aktiviert, dann werden alle zum entsprechenden Signalnetz gehörenden Objekte durch Highlight angezeigt. Ist eine Verbindung realisiert, dann wird die entsprechende Airline gelöscht. Beim manuellen Routen erzeugte Kurzschlüsse oder Abstandsverletzungen werden vom Online-Check sofort (durch Highlight bzw. Einrahmung) angezeigt.

Sollte es Ihnen im eingestellten Eingaberaster nicht gelingen, einen Pin anzuschließen (weil dieser in einem zu feinen Raster liegt), dann müssen Sie auf ein feineres Eingaberaster schalten, oder u.U. das Eingaberaster über das Menü Ansicht freigeben.

Sehr hilfreich für das interaktive Routen sind die Funktionen Zoom Fenster und Zoom Uebersicht aus dem Menü Ansicht. Mit diesen Funktionen können Sie abwechselnd ein definiertes Fenster herauszoomen oder in die Übersichtsdarstellung schalten.

Wichtig für das interaktive Routen ist die Wahl der Vorzugslage, die beim Aufruf des **Layouteditors** auf die Signallage 1 eingestellt ist. Die aktuell definierte Vorzugslage wird immer angewählt, wenn beim Pick eines Elements keine eindeutige Auswahl möglich ist, also z.B. beim Verlegen neuer Leiterbahnen von Lötaugen weg, die über alle Lagen definiert sind. Wenn Sie z.B. mit dem Verlegen neuer Leiterbahnen per Default auf der Bauteilseite beginnen wollen, dann empfiehlt es sich, die Vorzugslage über das Menü Ansicht entsprechend einzustellen.

Der Pickalgorithmus für Leiterbahnecken und Leiterbahnsegmente sorgt dafür, dass bei mehreren möglichen Pickelementen im Fangbereich das Leiterbahnelement mit dem minimalen Abstand zum Pickpunkt selektiert wird. Damit ist auch in Übersichtsdarstellungen ein gezieltes Anwählen von Leiterbahnen möglich. Außerdem ist der Fangbereich für die Leiterbahnanwahl unabhängig vom Zoomfaktor auf jeden Fall so groß wie die Leiterbreite, d.h. zur Selektion von Leiterbahnen ist es nicht notwendig, die Leiterbahnmitte exakt zu treffen (was bei breiten Leiterbahnen bzw. hohen Zoomfaktoren ggf. recht schwierig wäre).

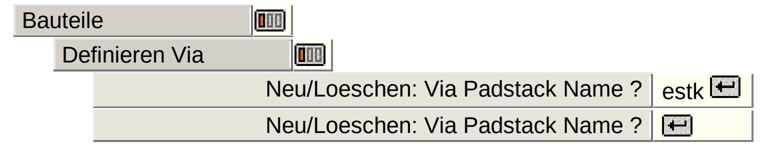
Highlight Netz

Die Funktion Highlight Netz im Menü Leiterbahnen bewirkt eine grafische Kennzeichnung aller zum selektierten Netz gehörenden Verbindungen durch eine spezielle Farbe ("Highlight"). Auf den Versorgungslagen werden die Kreise für Wärmefallen und Isolationen für das selektierte Netz ggf. ebenfalls gehighlighted dargestellt, und es ist auch möglich, isolierte Pins und Potentialflächen zum Highlight zu selektieren. Beim Verlegen neuer Leiterbahnen mit dem Startpunkt auf einer passiven Kupferfläche wird ein Highlight für die an der Kupferfläche angeschlossenen Netze ausgelöst. Ein aktiviertes Netzhighlight kann durch abermaliges Selektieren des Netzes oder eines zugehörigen Pins oder Potentialbereichs über die Funktion Highlight Netz wieder zurückgesetzt werden. Nach Aktivierung der Funktion Highlight Netz stehen neben der Standardoption Highlight Netze zusätz;ich die Optionen Colorieren Netze zum Einfärben von Netzen und Alle ruecksetzen zur Deaktivierung sämtlicher Netzhighlights zur Auswahl.

In **BAE HighEnd** bewirkt die Funktion Highlight Netz ein Highlight bzw. eine Highlight-Rücknahme der selektierten Netze in *allen* aktuell geladenen Plänen der aktuellen Projektdatei auf Layout- und Schaltplanebene (globales Netz-Highlight, Cross-Probing).

Selektion der Durchkontaktierung

Für Durchkontaktierungen (Vias, Umsteiger beim Signallagenwechsel) wird (zumindest) ein Padstacksymbol benötigt. Die dazu nötige Viazuweisung muss *vor* dem Verlegen der Leiterbahnen durchgeführt werden. Definieren Sie das Padstacksymbol **estk** als Via:



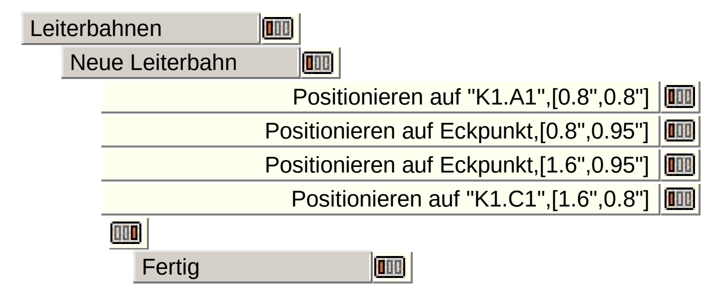
Das Padstacksymbol **estk** ist als Pinsymbol im bereits platzierten Bauteil **v1000** (Gehäuse **d04a25**) verwendet und somit in der aktuellen Projektdatei direkt verfügbar. In der mit der BAE-Software ausgelieferten Layoutbibliothek sind einige Via-Definitionen enthalten, von denen üblicherweise der Padstack **via** als Standard-Via verwendet werden kann. Ist kein Via definiert, dann gibt die Funktion Neue Leiterbahn die Meldung **Die Standardvia Definition fehlt!** aus. Es ist möglich, mehrere Vias (d.h. auch Blind und Buried Vias) zu definieren. Im Laufe der Bearbeitung lassen sich die Viazuweisungen beliebig ändern. Für alle neu zu verlegenden Leiterbahnen werden die zuletzt definierten Vias benutzt. Bereits platzierte Vias werden durch die Viazuweisung nicht geändert. Bei einem Lagenwechsel wird jeweils automatisch das Via mit der geringstmöglichen Lagenbelegung gesetzt. Die Viazuweisungen sind auch im **Autorouter** wirksam, d.h. der **Autorouter** legt alle Leiterbahnen unter Benutzung der zuletzt zugewiesenen Vias. Ausgenommen hiervon sind lediglich nicht vom **Autorouter** zu bearbeitende, d.h. fixierte Leiterbahnen.

Standardbreiten

Einige Untermenüs der Leiterbahn-Funktionen erlauben während des Verlegens von Leiterbahnen das Umschalten zwischen schmaler und breiter Leiterbahnführung ("Necking", "Bending"). Die Standardwerte für diese Leiterbreiten können im Menü Einstellungen über die Funktion Standardbreiten (Schmal, Default 0.3mm; Breit, Default 1.00mm) eingestellt werden. Beim Verlegen einer neuen Leiterbahn wird immer zunächst der für Schmal eingetragene Wert verwendet.

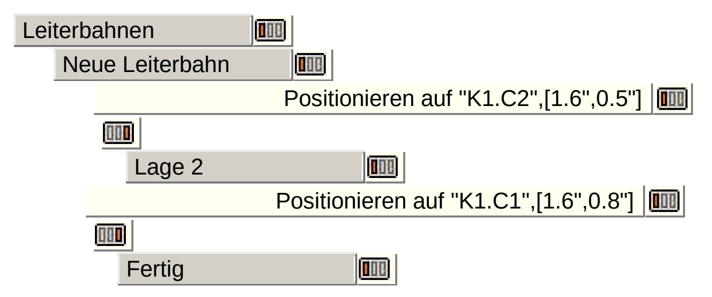
Manuelles Verlegen von Leiterbahnen

Verbinden Sie mit den folgenden Kommandos die zum Netz **Vdd** gehörenden Anschlüsse **A1** und **C1** des Bauteils **K1** miteinander:

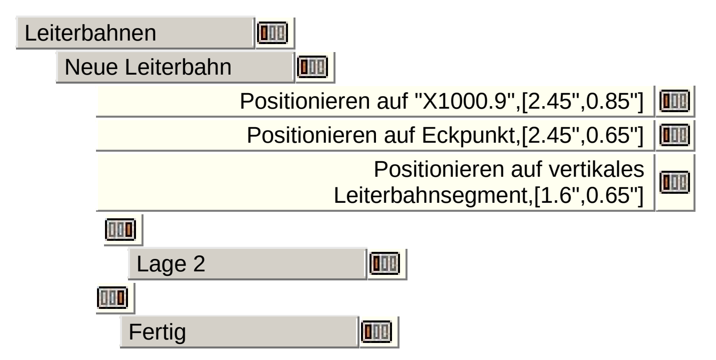


Leiterbahneckpunkte werden immer durch Betätigung der linken Maustaste gesetzt. Mit der rechten Maustaste gelangt man in ein Untermenü, in dem über die Option Fertig der Endpunkt der Leiterbahn gesetzt wird. Dieses Untermenü, das auch während der Änderung von Leiterbahnen (Ecke einfuegen, Segment bewegen, ...) verfügbar ist, ermöglicht auch den Sprung auf eine andere Signallage, das Einstellen einer anderen Leiterbahnbreite, die Erzeugung kreisbogenförmiger Leiterbahnsegmente, Relativ- und Absolutsprünge, usw.

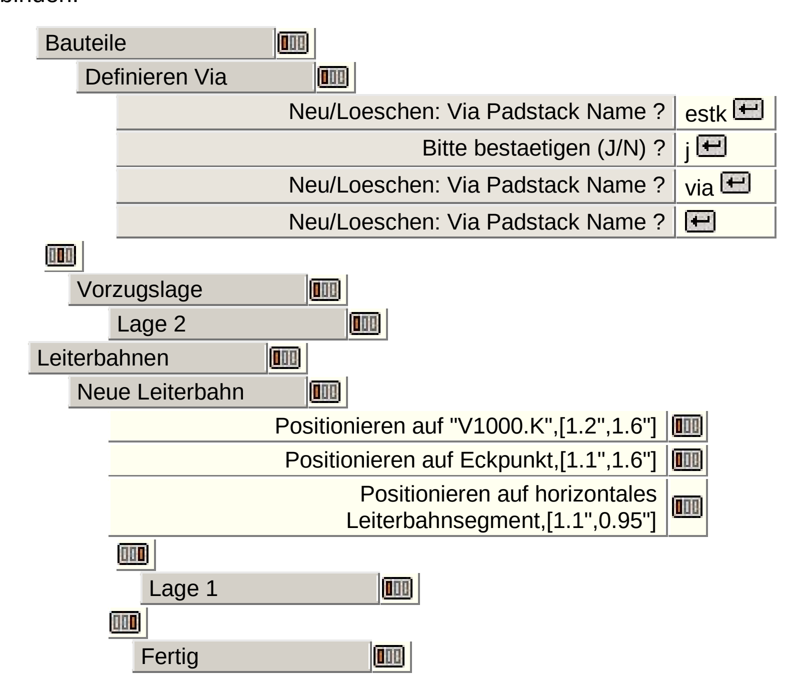
Verbinden Sie die beiden Anschlüsse **C2** und **C1** des Bauteils **K1** durch ein Leiterbahnsegment auf der Signallage 2 miteinander:



Legen Sie nun vom Pin **9** des Steckers **X1000** auf der Signallage 1 ein Leiterbahnsegment, das Sie über ein Via an die zuvor gelegte Leiterbahn anschließen:



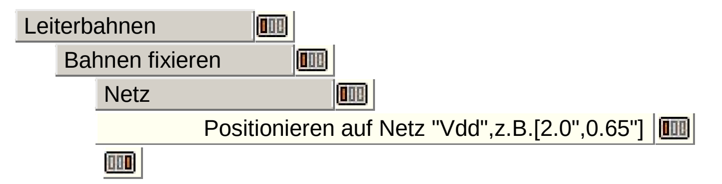
Deselektieren Sie nun das Via **estk**, selektieren Sie statt dessen das (in Kapitel 4.2.2 in der Datei **demo.ddb** erstellte) Padstacksymbol **via** als Durchkontaktierung, und legen Sie auf Signallage 2 eine Leiterbahn vom Pin **K** der Diode **V1000**, die Sie mit dem zuvor selektierten Via mit der horizontalen Leiterbahn zwischen den Pins **A1** und **C1** des Relais-Bauteils **K1** verbinden:



Legen Sie nun zur Übung nach Belieben weitere Leiterbahnen. Experimentieren Sie dabei auch mit den Funktionen zum Bewegen, Teilen und Löschen von Segmenten, zum Einfügen, Bewegen und Löschen von Leiterbahneckpunkten, zum Ändern von Leiterbahnbreiten, zur Erzeugung kreisbogenförmiger Leiterbahnen, usw. Im über die rechte Maustaste erreichbaren Kontextmenü der Funktion Segment bewegen stehen die Spezialoptionen Ohne Nachbarn bewegen, Mit Nachbarn bewegen und Nachbarn anpassen. zur Verfügung. Bei der Voreinstellung Ohne Nachbarn bewegen werden nur die Endpunkte des Leiterbahnsegments bewegt. Mit der Option Mit Nachbarn bewegen hingegen werden (soweit dies zur Beibehaltung der bisherigen Segmentabknickwinkel nötig ist) auch die angeschlossenen Segmente nachgeführt. Bei Aktivierung von Nachbarn anpassen. werden Nachbarsegmente soweit möglich entsprechend dem Schnittpunkt von neuem Segment und Nachbarsegment angepasst. Dieser Arbeitsmodus ist insbesondere für die Verschiebung von Diagonalsegmenten an Leiterbahnknickstellen nützlich. Ist eine Anpassung der benachbarten Segment nicht möglich, dann wird automatisch die Option Ohne Nachbarn bew. angewendet. Bedienen Sie sich beim Experimentieren mit den Prozeduren zur Leiterbahnbearbeitung der Funktionen Undo und Redo, um Realisierungsalternativen durchzuspielen.

Fixieren von Leiterbahnen

Dürfen (vorverlegte) Leiterbahnen vom **Autorouter** nicht verändert werden, dann müssen diese fixiert werden. Fixieren Sie mit folgenden Kommandos die vorverlegten Bahnen des Signalnetzes **Vdd**:



Nach dem Aufruf der Fixier-Funktion werden alle aktuell fixierten Leiterbahnen durch Highlight angezeigt. Beim Verlassen dieser Funktion wird dieses spezifische Highlight automatisch zurückgesetzt. Mit der Funktion Bahnen freigeben können fixierte Leiterbahnen wieder freigegeben werden.

Die Funktionen zur manuellen Bearbeitung von Leiterbahnen haben keine Einfluss auf aktuell gesetzte Fixiert-Attribute der bearbeiteten Leiterbahnelemente und Durchkontaktierungen. Das bedeutet, dass nach einer manuellen Nachbearbeitung vorher fixierter Leiterbahnen keine Notwendigkeit einer neuerlichen Fixierung besteht, um in einem nachfolgenden **Autorouter**-Lauf die Umverlegung (oder gar Herausnahme) solcher vorverlegten Leiterbahnen zu unterdrücken.

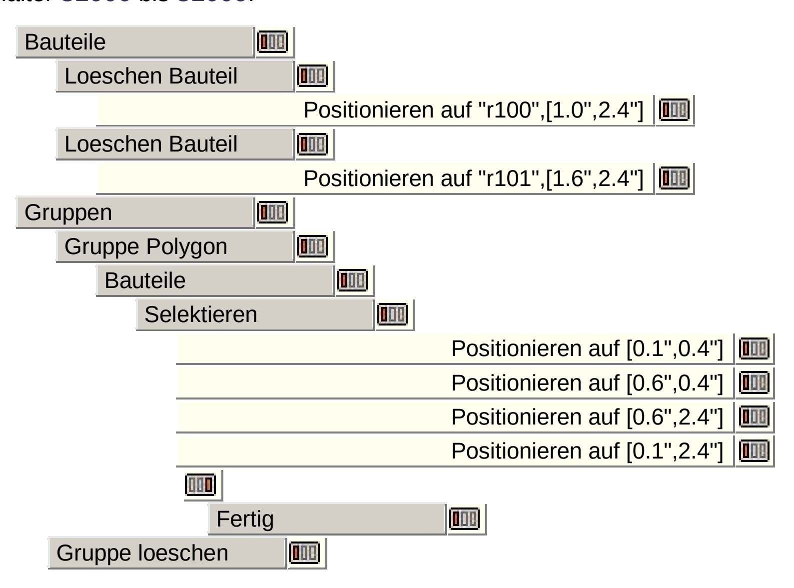
4.4 Autoplacement

Das BAE-Layoutsystem ist mit mächtigen **Autoplacement**-Verfahren ausgestattet. Vor der automatischen Platzierung können die zu platzierenden Bauteile nach dem Mengenprinzip selektiert werden. Mit dem Matrixplacement-Verfahren kann eine selektierbare Menge von Bauteilen automatisch auf einem definierbaren Einbauplatzraster platziert werden. Die integrierten Initialplacement-Funktionen ermöglichen die vollautomatische Durchführung der Bauteilplatzierung. Die Platzierungsoptimierung bietet Funktionen zum automatischen Bauteil- und Pin-/Gattertausch an.

Die **Autoplacement**-Funktionen sind im Menü Bauteile des **Layouteditors** unter Bauteilmenge, Autoplacement und Matrixplacement untergebracht.

Sofern Sie sich nicht im **Layouteditor** befinden, sollten Sie diesen zunächst aufrufen und das Layout `board` aus der Datei `demo.ddb` laden.

Löschen Sie nun mit den folgenden Kommandos die Widerstände `r100` und `r101` sowie (mit der Hilfe der Gruppenfunktion) die Schalter `s1000` bis `s1009`:

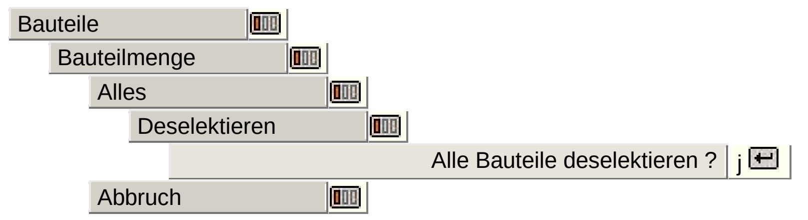


In den folgenden Abschnitten werden die soeben gelöschten Bauteile mit Hilfe von **Autoplacement**-Funktionen wieder auf das Layout zurückplatziert.

4.4.1 Bauteilmenge

Über Bauteilmenge können die zu platzierenden Bauteile ausgewählt werden. Dabei ist die Selektion und Deselektion über Bibliotheksteilnamen, Bauteilnamen und Wildcards möglich.

Deselektieren Sie zunächst mit den folgenden Kommandos alle unplatzierten Bauteile:



Werfen Sie mit den folgenden Kommandos einen Blick auf die Bauteilliste:



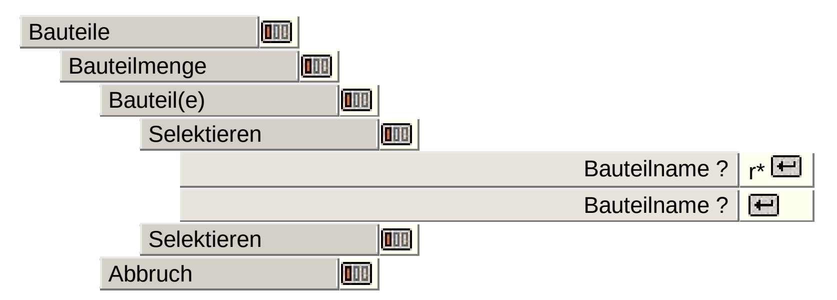
Das System sollte nun die folgende Liste mit allen in der Netzliste definierten Bauteilen auf dem Bildschirm ausgeben:

```
Datei : demo.ddb  Plan : board  Bauteile : 23

 : c100   chip1210  P  : c101   chip1206  P  : ic10   dil14     P
 : k1     relais    P  : r100   r04a25    U  : r101   r04a25    U
 : r102   r04a25    P  : r103   r04a25    P  : r104   minimelf  P
 : r105   chip1206  P  : s1000  s1dilo    U  : s1001  s1dilo    U
 : s1002  s1dilo    U  : s1003  s1dilo    U  : s1004  s1dilo    U
 : s1005  s1dilo    U  : s1006  s1dilo    U  : s1007  s1dilo    U
 : s1008  s1dilo    U  : s1009  s1dilo    U  : v1     to92      P
 : v1000  d04a25    P  : x1000  xsubd9bl  P  - Ende -
```

Die Bauteilliste enthält alle in der Netzliste definierten Bauteile mit Bauteilname, Bibliotheksteilname und dem Hinweis, ob das Bauteil bereits platziert ist (**P**), oder ob es noch unplatziert ist (**U**). Der Eintrag **S** bedeutet, dass das betreffende, unplatzierte Bauteil für die Platzierung selektiert ist. Durch die Eingabe von A und Betätigen der Eingabetaste ↵ gelangen Sie wieder in das Menü.

Selektieren Sie mit den folgenden Kommandos alle Bauteile, deren Namen mit **r** beginnen, zur Menge der zu platzierenden Bauteile:



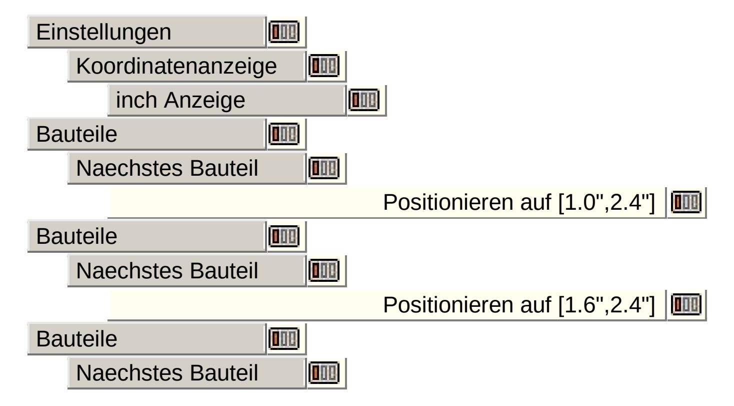
Die Auswahl von Bauteilnamen für die Bauteilmenge mit den Optionen Selektieren und Deselektieren der Funktion Bauteil(e) aus dem Menü Bauteilmenge kann auch durch Mausselektion im angebotenen Popupmenü vorgenommen werden. Wahlweise kann ein Bauteilnamensmuster per Tastatur eingegeben werden. Im Auswahlmenü sind jeweils nur die unselektierten, bzw. selektierten Bauteile aufgelistet.

Mit obigen Kommandos wurden die beiden noch nicht platzierten Widerstände **r100** und **r101** für die Platzierung selektiert. Sie können dies mit Hilfe der Funktion Bauteilliste überprüfen (Eintrag **S** für diese beiden Bauteile).

Mit der Funktion Alles aus dem Untermenü Bauteilmenge können alle noch nicht platzierten Bauteile für für die Platzierung selektiert werden. Mit der Funktion Macro(s) besteht die Möglichkeit der Selektion über den Bibliotheksnamen (z.B. **`so16`** für alle unplatzierten Bauteile mit der Gehäusebauform SO16, oder **`dil*`** für alle unplatzierten DIL-Gehäuse). Mit der jeweiligen Unterfunktion Deselektieren können Bauteile selbstverständlich auch wieder aus der Menge der zu platzierenden Bauteile entfernt werden.

Das Untermenü Bauteilmenge bietet darüber hinaus mit Blockliste und Block auch Funktionen zur selektiven Auswahl von Bauteilen, die in einem speziellen Blockschaltbild eines hierarchischen Schaltplanentwurfs definiert sind. Hierbei erfolgt die Selektion oder Deselektion der zu platzierenden Bauteile jeweils durch die Spezifikation des Blocknamens. Dieser Blockname wird bei hierarchischen Designs über das Attribut **`$blkname`** automatisch durch den **Packager** in den betroffenen Bauteilen eingetragen.

Im Kapitel 4.3.2 (Abschnitt Manuelle Platzierung) wurde bereits auf die Berücksichtigung der Bauteilmenge durch die Funktionen Neues Bauteil und Naechstes Bauteil hingewiesen. Stellen Sie mit den folgenden Kommandos die Koordinatenanzeige auf Inch ein, und platzieren Sie mit Hilfe der Funktion Naechstes Bauteil alle in der Bauteilmenge zur Platzierung selektierten Bauteile (es sind dies die Widerstände **`r100`** und **`r101`**):



Ist die Bauteilmenge leer (weil alle zuvor selektierten Bauteile platziert wurden), dann gibt die Funktion Naechstes Bauteil die Meldung

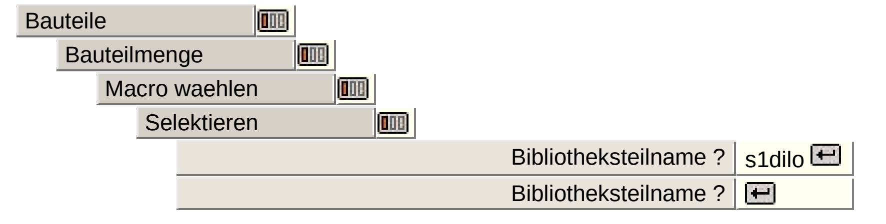
```
Alle selektierten Bauteile sind bereits platziert!
```

aus.

4.4.2 Matrixplacement

Das Matrixplacement-Verfahren ist ein Initialplacement-Algorithmus, mit dessen Hilfe sich die in der Bauteilmenge enthaltenen Bauteile auf in Matrixform definierten Einbauplätzen automatisch platzieren lassen.

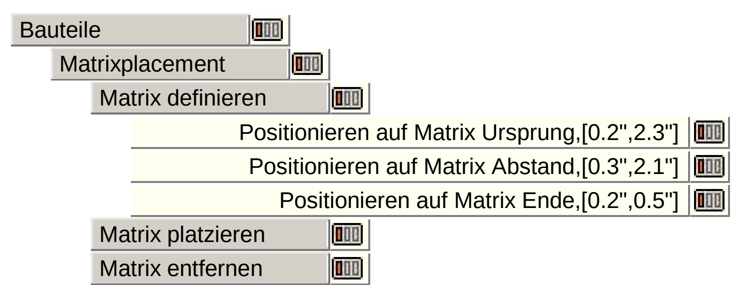
Selektieren Sie mit den folgenden Kommandos alle Bauteile mit dem Bibliotheksteilnamen **`s1dilo`** (dies sind die Schalter **`s1000`** bis **`s1009`**) zur Bauteilmenge:



Stellen Sie den Standardwinkel für die Platzierung auf 270 Grad (ohne Spiegelung) ein (die Option hierfür ist ggf. im Bereich Bauteile des Dialogs Einstellungen aus dem Menü Einstellungen erreichbar):



Definieren Sie nun eine Platzierungsmatrix, Platzieren Sie die in der Bauteilmenge enthaltenen Bauteile auf die Matrix, und entfernen Sie die Matrix wieder:



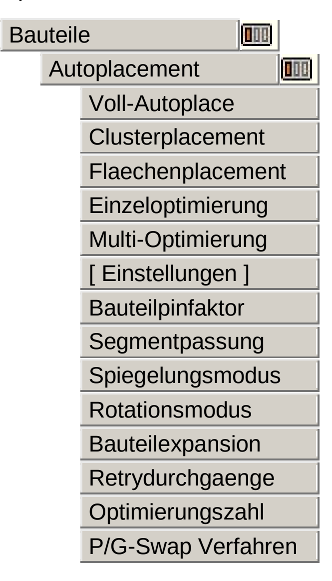
Die Funktion Matrix platzieren platziert alle in der Bauteilmenge enthaltenen Bauteile auf die definierte Platzierungsmatrix, wobei auch der aktuell eingestellte Platzierungswinkel und ggf. der Spiegelungsmodus für die Platzierung mit berücksichtigt werden.

Es sind nun alle in der Netzliste definierten Bauteile platziert. Sie können dies mit der Funktion Bauteilliste aus dem Untermenü Bauteilmenge überprüfen.

4.4.3 Initialplacement

Die im System integrierten Initialplacement-Funktionen ermöglichen die vollautomatische Durchführung der Bauteilplatzierung. Hierbei werden die unplatzierten Bauteile innerhalb der Platinenumrandung auf dem aktuell eingestellten Platzierungsraster platziert, wobei vorplatzierte Bauteile (Stecker, LEDs, etc.) ebenso berücksichtigt werden wie die Vorgaben aus der Netzliste. Abblockkondensatoren und SMD-Bauteile werden automatisch erkannt. Die Lötseite kann für die SMD-Platzierung wahlweise gesperrt oder freigegeben werden. Die Bauteile werden selbsttätig in 90-Grad-Schritten gedreht, wobei sich die Freiheitsgrade bei der Bauteilrotation für eine fehlersichere Bestückung wahlweise einschränken lassen. Optional kann ein Bauteilexpansionsparameter zur Generierung von Freiflächen zwischen den Bauteilen definiert werden. Die automatische Platzierung wird durch einstellbare Gewichtungsfaktoren zur Berücksichtigung von Netzlistenvorgaben und zur Bewertung der Bauteil-Segmentpassung gesteuert. Bereits während der Platzierung werden Rip-Up/Retry-Läufe eingeschoben, um die Ausnutzung der Platinenfläche zu optimieren.

Die Initialplacement-Funktionen sind im Untermenü Autoplacement integriert, welches die folgenden Funktionen umfasst:



Zur automatischen Bauteilplatzierung stehen die Funktionen Voll-Autoplace, Clusterplacement und Flaechenplacement zur Verfügung. Es werden jeweils die aktuell zur Bauteilmenge selektierten Bauteile platziert (siehe Kapitel 4.4.1). Damit lassen sich selektierbare Bauteilgruppen automatisch platzieren. Bereits platzierte Bauteile, die nicht in der Bauteilmenge enthalten sind, werden von den Platzierungsroutinen wie fixierte Bauteile behandelt und als solche nicht umplatziert.

Das erste für die Platzierung gewählte Bauteil wird auf einem Startpunkt positioniert, der durch die Funktion Voll-Autoplace automatisch ermittelt wird bzw. in den Funktionen Clusterplacement und Flächenplacement vom Anwender interaktiv festzulegen ist (Prompt **Startpunkt fuer Platzierung waehlen!**). Die Auswahl des Startpunkts ist von essentieller Bedeutung für die Qualität des Platzierungsergebnisses, da sich die Platzierung von Folgebauteilen jeweils an der Platzierung des ersten Bauteils ausrichtet. Wenn das erste zu platzierende Bauteil aus Platzmangel nicht am Startpunkt platziert werden kann, dann ist auch keines der Folgebauteile platzierbar. Wird ein Startpunkt außerhalb der Platinenumrandung gewählt, dann ist kein Bauteil platzierbar. Ist keine Platinenumrandung definiert, dann kann der Startpunkt frei gewählt werden, und Bauteile werden u.U. auch außerhalb der aktuell definierten Elementgrenzen platziert.

Der Fortgang der Bearbeitung während der automatischen Platzierung wird in der Statuszeile protokolliert (Anzeige **Pass : 1/1 Bauteil : <b>/<n>**). Dieser Prozess kann per Tastendruck abgebrochen werden, was ggf. mit der Meldung **Autoplace abgebrochen!** quittiert wird. Dabei ist zu beachten, dass es vor dem tatsächlichen Funktionsabbruch u.U. zu kurzen Wartezeiten kommen kann, da der aktuell aktive Ripup- und Retry-Pass in jedem Fall komplettiert werden muss.

Die erfolgreiche Beendigung der Initialplacement-Funktion wird durch die Meldung **Es wurden keine Fehler festgestellt.** angezeigt, was zugleich auch bedeutet, dass alle Bauteile platziert werden konnten. Eine durch die Platzierungsfunktion ausgegebene Meldung der Form **<n> Bauteile konnten nicht platziert werden!** gibt die Anzahl **<n>** der Bauteile an, die mit den eingestelltem Parametern nicht auf dem zur Verfügung stehenden Platz platziert werden konnten. In diesem Fall sind ggf. andere Parameter für die Platzierung (feineres Platzierungsraster, Reduzierung der Bauteilexpansion, Freigabe Bauteildrehung bzw. SMD-Spiegelung; Parametereinstellung siehe unten) zu wählen, um (nach einem Undo) in einem neuerlichen Platzierungslauf eine Komplettplatzierung zu ermöglichen.

Voll-Autoplace

Die Funktion Voll-Autoplace führt ein Clusterplacement (siehe unten) mit automatisch gewähltem Startpunkt für die Platzierung sowie anschließend eine (multi-pass) Platzierungs-Optimierung (siehe unten, Kapitel 4.4.4) mit den dafür eingestellten Parametern für Optimierungszahl und Pin/Gate-Swap-Verfahren durch. Der Startpunkt für die Platzierung ergibt sich dabei aus dem Schwerpunkt der Platinenumrandung. Liegt dieser Schwerpunkt nicht innerhalb der Platinenumrandung, dann können mit Voll-Autoplace keine Bauteile platziert werden, und es muss Clusterplacement oder Flächenplacement anstelle Voll-Autoplace verwendet werden.

Ist keine Platinenumrandung definiert, dann wird der absolute Nullpunkt des aktuell geladenen Layouts als Startpunkt gewählt. In diesem Fall sollte nach Ablauf der Platzierung auf die Übersichtsdarstellung umgeschaltet werden, da anschließend Bauteile u.U. außerhalb der aktuell definierten bzw. sichtbaren Elementgrenzen platziert sind.

Clusterplacement

Beim Clusterplacement ist zunächst der Startpunkt für die Platzierung zu wählen. Dieser Startpunkt wird zur Positionierung des ersten für die Platzierung ausgewählten Bauteils verwendet.

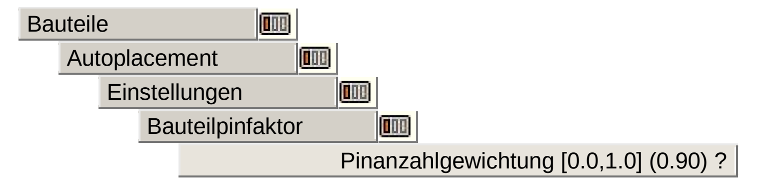
Die Funktion Clusterplacement wertet die Netzliste aus und nimmt eine Analyse und Klassifizierung der vorkommenden Netze und Bauteile vor. Danach werden Gruppen von Bauteilen zu Clustern zusammengefasst, die dann entsprechend der für die einzelnen Bauteilcluster ermittelten Platzierungsprioritäten nacheinander auf dem Layout platziert werden. Zur Clusterbildung werden jedem Bauteil mit mehreren Anschlüssen die jeweils verbundenen Bauteile mit einer geringen Anzahl von Anschlüssen (weniger als vier) zugeordnet. Bauteile die ausschließlich zu als Versorgungsnetzen klassifizierten Netzen Verbindungen haben (dies sind in aller Regel Abblockkondensatoren), werden zunächst nicht berücksichtigt und am Ende auf die mit den Versorgungsnetzen verbundenen Gruppen aufgeteilt. Dadurch ergibt sich anschließend automatisch eine sinnvolle Platzierung der Abblockkondensatoren in der Nähe der zugehörigen ICs. Die Aufteilung der Abblockkondensatoren erfolgt nach Alphabet. Da nicht zwischen Steckern und normalen Bauteilen unterschieden werden kann, ist es von Vorteil, bei der Namensgebung für die Bauteile die Stecker am Ende des Alphabets anzusiedeln (z.B. Namensmuster **`x???`** für Stecker und **`ic???`** für integrierte Schaltungen), um zu verhindern, dass Abblockkondensatoren an Stecker zugewiesen werden.

Flaechenplacement

Die Funktion Flaechenplacement führt eine Flächenplatzierung durch. Dabei ist zunächst der Startpunkt für die Platzierung zu wählen. Flächenplacement bedeutet, dass die Netzlistendaten nicht berücksichtigt werden, sondern dass lediglich die Bauteilabmessungen als Kriterium für die Platzierung herangezogen werden. Mit dieser Funktion lässt sich zum Beispiel leicht überprüfen, ob der auf der Leiterkarte zur Verfügung stehende Platz überhaupt ausreicht.

Bauteilpinfaktor

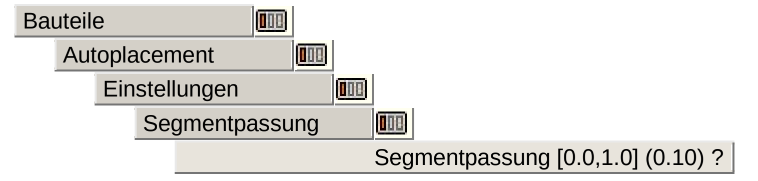
Der Bauteilpinfaktor steuert die Reihenfolge, in der die Bauteilgruppen platziert werden. Für die Auswahl des nächsten zu platzierenden Bauteils steht die Kombination zweier Strategien zur Verfügung. Die erste Strategie wählt einfach das Bauteil mit der größten Anzahl von Verbindungen zu den bereits platzierten Bauteilen aus, während die zweite Strategie das Bauteil auswählt, bei dem das Verhältnis der Pins mit Verbindungen zu bereits platzierten Bauteilen zur Gesamtzahl seiner Anschlüsse am höchsten ist. Über den Bauteilpinfaktor wird gesteuert welche Strategie höher zu bewerten ist. Mit der Funktion



können für den Bauteilpinfaktor Werte im Bereich von 0.0 (reine Bewertung der Verbindungsanzahl) bis zu 1.0 (reine Bewertung des Prozentsatzes verbundener Anschlüsse) angegeben werden. Der Defaultwert für den Bauteilpinfaktor beträgt 0.9. Von der Verteilung der Netze her sind hohe Bauteilpinfaktoren grundsätzlich besser als niedrige Faktoren. Allerdings führen hohe Bauteilpinfaktoren häufig dazu, dass besonders bei kleinem Platzangebot zu Beginn der Platzierung viele kleine Bauteile den vorhandenen Platz so fragmentiert belegen, dass später zu platzierende große Bauteile keinen Platz mehr finden.

Segmentpassung

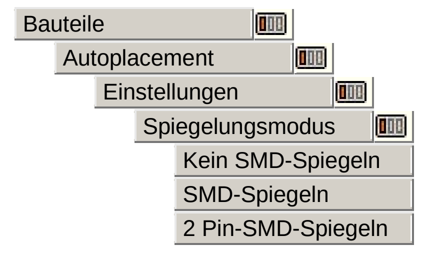
Die Segmentpassung gibt an, inwieweit die Passung der Kantenlängen benachbarter Bauteile berücksichtigt wird, d.h. ob Bauteile gleicher Größe bevorzugt nebeneinander platziert werden. Mit der Funktion



wird der Bewertungsfaktor für die Segmentpassung festgelegt. Es können Werte von 0.0 (keine Berücksichtigung der Segmentpassung) bis 1.0 (Segmentpassung wird vergleichbar den Netzausdehungen bewertet) angegeben werden. Der Defaultwert für die Segmentpassung beträgt 0.1. Mit hoher Segmentpassung platzierte Layouts sehen in der Regel optisch ansprechender aus und sind bei Vorhandensein vieler Busverbindungen besser routbar. Layouts mit mehr "zufällig" verteilten Verbindungen und beengten Verhältnissen sind dagegen häufig bei einer Platzierung mit geringer Segmentpassung einfacher entflechtbar.

Spiegelungsmodus

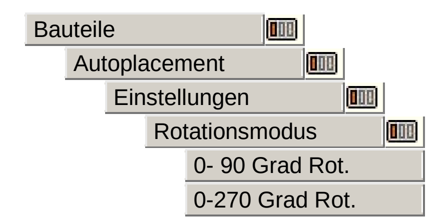
Der Spiegelungsmodus ist nur für die Platzierung von SMD-Bauteilen von Bedeutung. Bei der Voreinstellung Kein SMD-Spiegeln werden SMD-Bauteile nur ungespiegelt platziert. Mit der Einstellung SMD-Spiegeln kann die automatische Platzierung wahlweise auf der Ober- bzw. Unterseite durchgeführt werden. Die Option 2 Pin-SMD-Spiegeln dient der Einschränkung der automatischen Bauteilspiegelung auf SMDs mit nicht mehr als 2 Pins und erlaubt somit die Platzierung von Abblockkondensatoren und anderen Kleinbauteilen auf der Lötseite, während SMD-Bauteile mit mehr als zwei Pins in jedem Fall auf der Bauteilseite platziert werden. Mit der Funktion



kann die gewünschte SMD-Spiegelungsoption selektiert werden.

Rotationsmodus

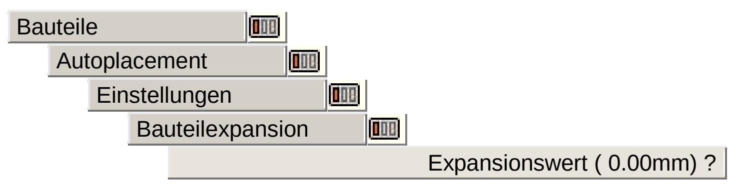
Der Rotationsmodus gibt an um welche Winkel die Bauteile bei der automatischen Platzierung gedreht werden dürfen. Die Standardeinstellung für den Rotationsmodus ist 0-270 Grad Rotation, d.h. die Bauteile dürfen in beliebigen 90 Grad-Schritten platziert werden. Mit der Option 0- 90 Grad Rotation werden die Bauteile nur ungedreht oder um 90 Grad nach links gedreht platziert. Dieser Modus kann angewendet werden um eine fehlersicherere Bestückung zu erreichen; außerdem verläuft damit die automatische Platzierung schneller, da weniger Platzierungsmöglichkeiten geprüft werden müssen. Mit der Funktion



kann zwischen den beiden Rotationsoptionen gewählt werden.

Bauteilgröße, Bauteilexpansion, Abblockkondensatoren, Platzierungsraster

Die Bauteile werden im aktuell eingestellten Eingaberaster platziert. Dieses Platzierungsraster sollte also vor Durchführung des Initialplacement im Menü Ansicht über die Funktion Raster/Winkel festgelegt werden. Der Platzbedarf (d.h. die Größe) eines Bauteils wird durch die Elementgrenzen des zugehörigen Makros bestimmt. Beim Erstellen von Bauteilbibliotheken sollte daher darauf geachtet werden, dass die Elementgrenzen nicht unnötig größer als die tatsächlichen Bauteilabmessungen sind. Um in weniger dicht gepackten Designs Platz zwischen den Bauteilen zu schaffen, kann eine Expansionsdistanz angegeben werden, um die die Bauteilumrandung vergrößert werden soll. Mit der Funktion

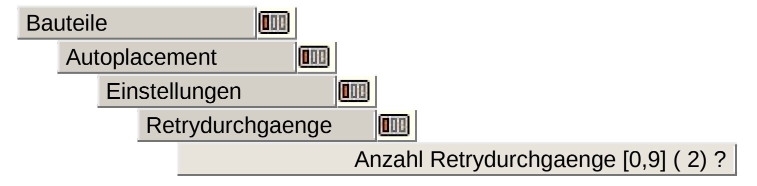


wird der Wert für die Bauteilexpansion eingestellt. Der Defaultwert hierfür beträgt 0.0mm (d.h. keine Bauteilexpansion). Der mit der Funktion Bauteilexpansion angegebene Expansionswert wird auf jeden Bauteiltyp ohne Berücksichtigung individueller Makrogrößen gleichermaßen angewendet. Ausgenommen hiervon sind Abblockkondensatoren, die automatisch erkannt werden und mit geringstmöglichem Abstand zu den zu versorgenden Bauteilen platziert werden. Nach Möglichkeit werden die Abblockkondensatoren hierbei je nach Orientierung der zu versorgenden ICs vorzugsweise jeweils rechts bzw. oberhalb der integrierten Schaltkreise platziert.

Im Vollautoplacement wird ein Automatismus zur schrittweisen Reduktion der Bauteilexpansion aktiviert, wenn keine vollständige Platzierung mit dem eingestellten Wert möglich ist. Der Wert für die Bauteilexpansion wird nach Möglichkeit solange um jeweils 25 Prozent reduziert bis entweder eine vollständige Platzierung möglich ist, oder der Wert für die Bauteilexpansion 0.2mm unterschreitet und auf Null gesetzt wird.

Retrydurchgänge

Da bei der Platzierung jeweils nur die Verbindungen zu bereits platzierten Bauteilen berücksichtigt werden können, kann sich im Laufe der Platzierung für ein bereits platziertes Bauteil eine optimalere Position ergeben. Damit solche Positionen auch genutzt werden, werden bei der automatischen Platzierung Retrydurchgänge eingeschoben, bei denen für jedes platzierte Bauteil nach einer besseren Position gesucht wird. In dieses Verfahren werden auch vorplatzierte, unfixierte Bauteile einbezogen. Vorplatzierte Bauteile, die nicht umplatziert werden dürfen, müssen daher vorher im **Layouteditor** (mit Hilfe der Gruppenfunktionen) fixiert werden. Mit der Funktion

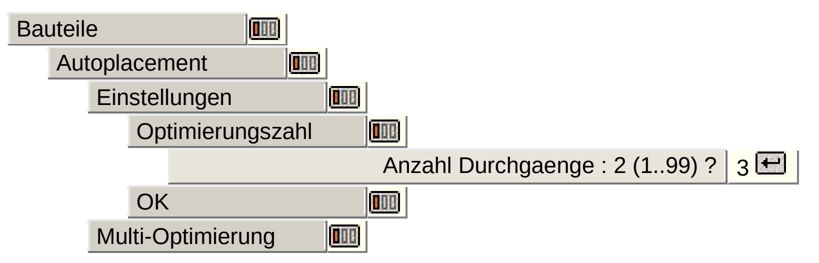


wird die Anzahl der Retrydurchgänge festgelegt. Die Voreinstellung beträgt zwei Durchgänge, d.h. es wird ein Retrydurchgang nach der Hälfte der platzierten Bauteile und ein Retrydurchgang am Ende der Platzierung durchgeführt. Es sollten nicht zu viele Retrydurchgänge angegeben werden, da dies einerseits stark in die Rechenzeit geht und andererseits bei der Umplatzierung von Bauteilgruppen Kleinbauteile (speziell Abblockkondensatoren) nicht immer an die neue Position mitgenommen werden können, wodurch sich für einzelne Verbindungen unverhältnismäßig lange Leiterbahnlängen ergeben können.

4.4.4 Platzierungsoptimierung

Die Platzierungsoptimierung ermöglicht den Aufruf eines Platzierungsalgorithmus, der iterativ Gehäuse gleicher Bauform gegeneinander vertauscht und Pin/Gate Swaps durchführt, um dadurch die Komplexität des Routingproblems zu entschärfen, d.h. eine Längenminimierung und Entflechtung der noch nicht realisierten Verbindungen ("Airlines", "Unroutes") herbeizuführen. Dieser Algorithmus kann einfach (Einzeloptimierung) oder mehrfach (Multi-Optimierung mit einstellbarer Optimierungszahl) durchlaufen werden, wobei mit der Funktion P/G-Swap Verfahren zwischen unterschiedlichen Swapverfahren gewählt werden kann; mit der Standardeinstellung Beide Swaps werden sowohl Bauteiltausch (Component Swap) als auch Pin-/Gattertausch durchgeführt, mit Nur Bauteil Swap wird der Pin/Gate Swap abgeschaltet, und mit Nur Pin/Gate Swap wird der Bauteiltausch abgeschaltet.

Führen Sie mit den folgenden Kommandos eine 3-fache Platzierungsoptimierung mit Bauteiltausch und Pin/Gate Swap durch:



Während der Optimierung zeigt das System in der Statuszeile den aktuellen Durchlauf und die Anzahl der bearbeiteten Bauteile an. Durch einen Tastendruck lässt sich die Platzierungoptimierung jederzeit abbrechen.

Beurteilen Sie mit Hilfe der Undo-Funktion das Resultat der Platzierungsoptimierung und beachten Sie dabei insbesondere die jeweilige Komplexität des "Airline-Geflechts".

Von der Platzierungsoptimierung ausgenommen sind fixierte Bauteile. Dürfen also bestimmte Bauteile (Stecker, Schalter, Leuchtdioden, usw.) durch die Platzierungsoptimierung nicht umplatziert werden, dann sind diese Bauteile vor dem Aufruf von **Autoplacement**-Funktionen (mit Hilfe der Gruppenfunktion) zu fixieren. Dasselbe gilt für Bauteile, für die kein Pin/Gate Swap durchgeführt werden darf (Relais, Mehrfach-Operationsverstärker, etc.). Wir weisen in diesem Zusammenhang auch mit Nachdruck darauf hin, dass ein automatischer Pin/Gate Swap nur dann durchgeführt werden darf, wenn alle in der logischen Bibliothek eingetragenen Swapdefinitionen auch wirklich zulässig sind. Ist dies nicht der Fall, oder bestehen darüber Zweifel, dann ist der automatische Pin-Gattertausch bei der Platzierungsoptimierung unbedingt mit Nur Bauteil Swap in der Funktion P/G-Swap Verfahren abzuschalten. Für Stecker darf in aller Regel kein Pin/Gate Swap definiert werden. Pin/Gate Swaps für Bauteile, bei denen spezielle Attributwerte (z.B. **`$val`** bei Widerstandsnetzwerken) gesetzt werden können, müssen als interne Swaps definiert sein, um zu verhindern, dass Gatter zwischen Bauteilen mit unterschiedlichen Werten getauscht werden. Die Definition von Pin/Gate Swaps erfolgt in der logischen Bibliothek mit Hilfe des Utilityprogramms **LOGLIB** (siehe hierzu auch die entsprechende Beschreibung im Kapitel 7.11 dieses Handbuchs).

4.5 Autorouter

Natürlich sollten Sie bei einem realen Projekt vor dem **Autorouter**-Aufruf die Versorgungslagen definieren bzw. kritische Leitungen oder den Versorgungskamm manuell vorverlegen und fixieren, ggf. die oberste Lage definieren, usw. Nach dem Autorouting sollte in jedem Fall ein Batch-Design Rule Check durchgeführt werden, bevor mit dem **CAM-Prozessor** die Fertigungsdaten erzeugt werden.

Die Benutzeroberfläche des **Bartels Autorouters** ähnelt der des **Layouteditors**. Der **Bartels Autorouter** unterstützt neben den eigentlichen Autorouting-Funktionen (Voll-Autorouter, Initial-Routing, Ripup- und Retry-Router, Fertigungsoptimierer) eine Reihe spezieller Platzierungs- und Routingfunktionen wie automatisches Initialplacement mit Platzierungsoptimierung, Einzelnetz- und Netzgruppen-Routing, Component-Routing, Bereichs- bzw. Blockrouting, Routen in gemischten Rastern, selektiver Bauteil- und Pin/Gate-Swap während des Rip-Up-Routings, usw. Die in **BAE HighEnd** integrierte Version des **Autorouters** bietet darüber hinaus mächtige Zusatzfunktionen basierend auf einer patentierten Technologie neuronaler Netzwerke. Der **Autorouter** der **BAE HighEnd**-Software bedient sich künstlicher Intelligenz zur automatischen Lösung spezieller Entflechtungsprobleme wie sie z.B. typischerweise beim Analogrouting oder bei der Erzeugung von Leiterbahnen mit speziellen elektrischen Eigenschaften bzw. bei der Generierung von Mikrowellenstrukturen auftreten. Hierzu arbeitet der **Neuronale Autorouter** mit Funktionen zur Erlernung und automatischen Anwendung von Regeln zur Lösung spezieller Entflechtungsprobleme und wird dabei zusätzlich noch unterstützt durch einen rasterlos arbeitenden, objektorientierten Routingalgorithmus mit integrierter Platzierungsoptimierung.

4.5.1 Programmaufruf

Der Aufruf des **Autorouters** erfolgt mit der Funktion Autorouter im Menü Datei des **Layouteditors**. Nach der Aktivierung dieser Funktion wird das aktuell geladene Layout gespeichert, und anschließend wird der **Autorouter** gestartet. War vorher im **Layouteditor** ein Layout geladen, so wird dieses auch automatisch im **Autorouter** geladen. Auf diesem Layout sollte (mit der Funktion Neue Umrandung im Menü Flaechen) eine Platinenumrandung definiert worden sein, und es sollte (mit der Funktion Definieren Via im Menü Bauteile des **Layouteditors**) eine Viazuweisung durchgeführt worden sein.

Beim Aufruf von **Autorouter**-Prozeduren können die folgenden Fehlermeldungen auftreten:

Die Platinenumrandung fehlt!	
Ursache:	keine Platinenumrandung definiert
Abhilfe:	**Layouteditor** - Flaechen - Neue Umrandung - ...
Die Standardvia Definition fehlt!	
Ursache:	keine Viazuweisung definiert
Abhilfe:	**Layouteditor** - Bauteile - Definieren Via - ...
Bauteile fehlen oder sind vom falschen Typ!	
Ursache:	noch nicht alle Netzlistenbauteile (mit der korrekten Gehäusebauform) platziert
Abhilfe:	ggf. **Layouteditor** - Bauteile - Update loeschen zur Elimination von mit falschen Gehäusebauformen platzierten Netzlistenbauteilen und anschließende interaktive bzw. automatische Platzierung der noch nicht platzierten Bauteile
Pin ausserhalb der Umrandung (<bauteilname>) !	
Ursache:	Netzlistenbauteil bzw. Netzlistenbauteilpin außerhalb der Platinenumrandung platziert
Abhilfe:	Korrektur der Bauteilplatzierung
Via ausserhalb der Umrandung!	
Ursache:	vorplatzierte, fixierte Durchkontaktierung(en) außerhalb der Platinenumrandung
Abhilfe:	Durchkontaktierung(en) außerhalb der Platinenumrandung freigeben bzw. löschen
Leiterbahn ausserhalb der Umrandung!	
Ursache:	vorverlegte, fixierte Leiterbahn(en) außerhalb der Platinenumrandung
Abhilfe:	Leiterbahn(en) außerhalb der Platinenumrandung freigeben bzw. löschen
Via Pad-Stack als Via ungeeignet!	
Ursache:	die Liste der Durchkontaktierungen enthält eine ungültige Via-Definition
Abhilfe:	Korrektur der Via-Definition(en); es muss zumindest ein Via über alle Signallagen definiert sein, und jedes Via muss eine Bohrung sowie Paddefinitionen für zumindest zwei adjazente Signallagen enthalten
Pad doppelt definiert!	
Ursache:	Layout enthält; mehrdeutige Bibliotheksdefinition(en); z.B. Pad(s) mit mehr als einer Kupferfläche oder Padstack(s) mit mehr als einem Pad auf derselben Lage
Abhilfe:	Pad- bzw. Padstack-Bibliotheksdefinition(en) korrigieren
Pad-Stack doppelt definiert!	
Ursache:	Layout enthält; mehrdeutige Bibliotheksdefinition(en); z.B. Bauteil(e) mit mehr als einem Padstack bzw. Pin an derselben Position
Abhilfe:	Bauteil-Bibliotheksdefinitionen korrigieren

Andere **Autorouter**-Fehlermeldungen werden ausgegeben, wenn das Layout Kurzschlüsse (verursacht durch vorverlegte, fixierte Leiterbahnen) enthält.

4.5.2 Hauptmenü

In der Benutzeroberfläche des **Autorouters** werden neben bereits aus dem **Layouteditor** bekannten Menüs (wie Undo/Redo, Ansicht bzw. Bilddarstellung, Parameter, Diverse) Funktionen zur automatischen Platzierung, zur Vorgabe von Options- und Strategieparametern, zur Durchführung der automatischen Entflechtung, sowie zur Aktivierung spezieller Routingfunktionen angeboten. Nach dem Aufruf des **Autorouters** befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden des **Autorouters** ist das Menü Diverse aktiviert, und der grüne Menübalken steht auf der Funktion Laden.

Unter Windows und Motif kann anstelle der Standard- bzw. Seitenmenükonfiguration wahlweise auch ein Benutzerinterface mit Pulldownmenüs aktiviert werden. Hierzu ist mit Hilfe des Utilityprogramms **BSETUP** das Kommando **WINMENUMODE** mit der Option **PULLDOWN** in das Setup der BAE-Software einzuspielen (siehe hierzu auch Kapitel 7.2). Bei der Verwendung von Pulldownmenüs ist das Hauptmenü als horizontal ausgerichtete Menüleiste am oberen Ende der Benutzerschnittstelle angeordnet.

Das Hauptmenü ist während der Dauer der Layoutbearbeitung mit dem **Autorouters** ständig verfügbar und ermöglicht die Aktivierung der folgenden Menüs:

Undo, Redo
Bilddarstellung
Preplacement
Autorouter
Interaktion
Optionen
Steuerung
Strategie
Parameter
Diverse

Im Menü Undo, Redo finden Sie die Undo-Funktion, mit der die letzten zwanzig Arbeitsschritte rückgängig gemacht werden können. Mit der Redo-Funktion kann der Undo-Befehl wieder aufgehoben werden. Diese Funktionen lassen sich insbesondere auch auf komplexe Operationen wie z.B. komplette **Autorouter**-Läufe anwenden.

Im Menü Ansicht bzw. Bilddarstellung, das Sie außer durch Selektion im Hauptmenü auch immer über die mittlere Maustaste erreichen können, können Sie Zoomfunktionen aktivieren, das Eingabe- bzw. Hintergrundraster definieren, oder die Farbtabelle einstellen.

Das Menü Preplacement des **Autorouters** entspricht dem Menü Autoplacement des Layouteditors (siehe Kapitel 4.4.3 und Kapitel 4.4.4). Damit stehen im **Autorouter** die aus dem Layouteditor bekannten Initialplacement-Funktionen sowie die Routinen zur Durchführung von Platzierungsoptimierungen zur Verfügung, d.h. im **Autorouter** kann (vor dem Start eines Autorouting-Prozesses) eine vollautomatische Vorplatzierung der Bauteile sowie eine Platzierungsoptimierung durch automatischen Bauteiltausch und Pin/Gate-Swap vorgenommen werden.

Das Menü Autorouter enthält die Funktionen zur Aktivierung der **Autorouter**-Prozeduren Voll-Autorouter, Optimierer und Einlesen Bahnen sowie die Funktionen Programm-Start und Programm-Setup, mit deren Hilfe ein festgelegter Ablauf unterschiedlicher Routerdurchläufe (Router-Passes) gestartet werden kann.

Das Menü Interaktion enthält die Funktionen zur Durchführung spezieller Routingoperationen wie z.B. Einzelnetzrouting, Routen von Netzgruppen, Component-Routing, Bereichsrouting, usw.

Das Menü Optionen dient dazu, die Optionsvorgaben für die nachfolgenden Routerläufe festzulegen. D.h. hier werden die grundlegenden Design- und Technologievorgaben (Lagenanzahl und Lagenzuordnung, Routingraster mit optionaler Benutzung des Halbrasters, Standard-Leiterbreite, Standard-Mindestabstand, maximal zulässige Viaanzahl, Via-Raster, Leiterbahnknicke on-grid oder off-grid, Pinanschlussverfahren) definiert. Werden grundlegende Optionsparameter (Lagenanzahl, Routingraster, Freigabe/Sperren des Halbrasters, Standard-Leiterbreite, Standard-Mindestabstand oder Pinanschlussverfahren) geändert, dann wird beim nachfolgenden Aufruf einer **Autorouter**-Prozedur (Voll-Autorouter, Initialrouting, SMD-Fanout-Routing, Rip-Up-Routing, Optimierer, Einlesen Bahnen) automatisch ein Router-Neustart mit Verwurf des aktuellen Routingergebnisses durchgeführt.

Das Menü Steuerung enthält wichtige Kontrollfunktionen zur Steuerung des Routerablaufs (Anzahl der Optimiererläufe, aktivieren/deaktivieren der netzübergreifenden Patternerkennung, Festlegung der Hartnäckigkeit des Rip-Up-Routers, SMD-Fanout-Router aktivieren/deaktivieren, automatisches Zwischenspeichern an- oder abschalten).

Über das Menü Strategie können die Strategieparameter und heuristischen Kostenfaktoren für die nachfolgenden Router- bzw. Optimiererläufe festgelegt werden (Art der Vorzugsrichtungs-Optimierung, Viakosten, Pinkanal-Kosten, Vorzugsrichtungskosten, Richtungsänderungskosten, Packungskosten, Kostenbasis für die statistische Leiterbahn-Verteilung, Bus-Abknickkosten, Rip-Up-Abstandskosten, Kreuzungskosten, Diagonalrouting-Kosten, Offgrid-Routing-Kosten).

Das Menü Parameter enthält Funktionen zur Definition des Zugriffspfades auf die Layoutbibliothek, zur Auswahl der Mincon-Funktion, zum Setzen des Koordinatenanzeigemodus, sowie zur Aktivierung der automatischen Datensicherung.

Im Menü Diverse kann der Programmabbruch, der Rücksprung in die Shell des **Bartels AutoEngineer** oder der Rücksprung in den **Layouteditor** veranlasst werden. Dieses Menü enthält weiterhin wichtige Dateiverwaltungsfunktionen zum Laden von Elementen und zum Auflisten von Dateiinhalten. Auch der explizite Aufruf von **User Language**-Programmen ist von diesem Menü aus möglich.

4.5.3 Modifizierte Benutzeroberfläche des Autorouters

Einige der mit der BAE-Software installierten **User Language**-Programme definieren implizite **User Language**-Programmaufrufe über die eine weit reichend modifizierte Benutzeroberfläche mit einer Vielzahl von Zusatzfunktionen (Startups, Toolbars, Menübelegung, Tastaturprogrammierung) aktiviert wird. Das **User Language**-Startupprogramm **BAE_ST** wird automatisch beim Aufruf des **Autorouters** gestartet. **BAE_ST** ruft seinerseits das **User Language**-Programm **UIFSETUP** auf, welches eine vordefinierte Menü- und Tastaturbelegung im **Autorouter** aktiviert. Änderungen bzw. Anpassungen der Menü- und Tastaturbelegung können *zentral* in der Quellcodedatei von **UIFSETUP** vorgenommen werden. Die aktuelle Tastaturbelegung kann mit dem **User Language**-Programm **HLPKEYS** angezeigt werden. Der Aufruf von **HLPKEYS** ist über die Funktion Tastaturbelegung aus dem Menü Hilfe möglich, sofern die vordefinierte Menübelegung aus **UIFSETUP** aktiviert ist. Mit dem **User Language**-Programm **UIFDUMP** kann die in der aktuellen Interpreterumgebung definierte Menü- und Tastaturbelegung in Form eines Reports angezeigt bzw. auf eine Datei ausgegeben werden. Mit dem **User Language**-Programm **UIFRESET** lässt sich die komplette Menü- und Tastaturbelegung zurücksetzen. **UIFSETUP**, **UIFDUMP** und **UIFRESET** sind auch über das Menü des **User Language**-Programms **KEYPROG** aufrufbar, welches zudem komfortable Funktionen zur Online-Tastaturprogrammierung sowie zur Verwaltung von Hilfstexten für **User Language**-Programme zur Verfügung stellt.

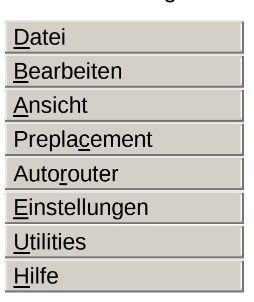
Die Windows- und Motifversionen des **Autorouters** ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Pulldownmenüs der Windows- und Motifversionen des **Autorouters** werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs ausgestattet. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

In den Windows- und Motifversionen des **Autorouters** sind die folgenden Dialoge für Parametereinstellungen implementiert:

- Einstellungen - Einstellungen: Allgemeine **Autorouter**-Parameter
- Ansicht - Einstellungen: Bilddarstellungsparameter
- Preplacement - Einstellungen: Parameter für automatische Platzierung
- Autorouter - Optionen: **Autorouter**-Optionen
- Autorouter - Steuerung: **Autorouter**-Steuerparameter
- Autorouter - Strategie: **Autorouter**-Strategieparameter
- Autorouter - Programm-Setup: **Autorouter**-Batcheinstellungen

In den Pulldownmenükonfigurationen werden die Standardfunktionen für Parametereinstellungen über das **User Language**-Programm **UIFSETUP** durch die obigen Menüfunktionen zum Aufruf der entsprechenden Dialoge ersetzt.

Bei der Verwendung von Pulldownmenüs unter Windows und Motif wird über das **User Language**-Programm **UIFSETUP** eine an Windows angepasste Menüanordnung mit zum Teil geänderten Funktionsbezeichnungen und einer Vielzahl von Zusatzfunktionen konfiguriert. Das Hauptmenü des **Autorouters** wird dabei wie folgt aufgebaut:



4.5.4 Grundsätzliches zur Bedienung

Automatische Parametersicherung

Im **Autorouter** sind Funktionen zur automatischen Sicherung wichtiger Design- und Bearbeitungsparameter implementiert. Die folgende Parameter werden automatisch beim Sichern des aktuell geladenen Layouts in der aktuell bearbeiteten Designdatei gespeichert:

- Zeitintervall für automatische Datensicherung
- Name der aktuell geladenen Layoutfarbtabelle
- Eingaberaster
- Hintergrundraster
- Raster- und Winkelfreigabe
- Koordinatenanzeigemodus
- Breitendarstellungswert
- Airlineanzeigemodus für Bauteilplatzierung
- Bibliothekszugriffspfad
- Mincon-Funktion

Die Elementnamen der zu sichernden Parametersätze werden vom aktuell bearbeiteten Layout abgeleitet, d.h. diese layoutspezifischen Parametersätze erhalten den Elementnamen des aktuell bearbeiteten Layouts. Beim Laden eines Layouts wird automatisch der entsprechende Parametersatz mitgeladen. Dadurch wird in komfortabler Weise eine spezifische Arbeitsumgebung zur Bearbeitung des selektierten Layouts aktiviert.

Grafikausgabe und Statusanzeigen

Nach dem Start des **Autorouters** wird der Fortgang des Routingprozesses sowohl grafisch als auch in einem Fenster mit statistischen Angaben angezeigt. Es ist moeglich, waehrend des Routingvorganges die Grafik- bzw. Bildschirmausgabe und die Aktualisierung der Statistik über den Routingprozess zu deaktivieren. Die Deaktivierung erfolgt durch Betätigung der Taste b. Bei deaktivierter Grafikausgabe wird die Meldung **`Router arbeitet. Aktivieren Bildschirmanzeige mit 'b'...`** angezeigt. Durch nochmaliges Betätigen der Taste 'b' wird die Grafikausgabe wieder aktiviert. Durch die Deaktivierung der Grafikausgabe lässt sich insbesondere in den Windowsversionen eine Beschleunigung des Routvorgangs um bis zu etwa 10 Prozent erreichen. Nach Beendigung des Routingprozesses erfolgt in jedem Fall ein Bildneuaufbau.

Während der Entflechtung zeigt der **Autorouter** in der Mitteilungszeile die Anzahl der vom Router bereits realisierten Verbindungen (in Relation zur Gesamtanzahl der Verbindungen) sowie die aktuelle Viaanzahl an. Gleichzeitig wird ein Statusfenster zur Anzeige von Informationen zum internen Ablauf der aktuell aktiven Routingprozedur angezeigt. Die vorletzte Zeile dieses Statusfensters enthält eine Kennung für den aktiven Routerpass (**`E`** - Einlesenen Bahnen, **`S`** - SMD-Viavorverlegen, **`I`** - Initialroutingpass, **`R`** - Rip-Up-Pass, **`P`** - Optimierer Patternsuchpass, **`O`** - Optimiererpass). Auf die Routerpass-Kennung folgt die Anzeige der Anzahl **`n`** der aktuell bearbeiteten Elemente in Relation zur Gesamtanzahl **`m`** der zu bearbeitenden Elemente in der Form **`n/m`**. Die Routerpass-Statuszeile wird abgeschlossen durch die Anzeige des **`c`** aktuellen Routerpasses in Relation zur Gesamtanzahl **`p`** der durchzuführenden Routerpasses in der Form **`c/p`**. Die angezeigten Werte sind kein zuverlässiges Maß für eventuell zu erwartende Gesamtroutzeiten, da die Bearbeitungszeit zur Realisierung einer Verbindung bzw. eines Netzes stark von der Komplexität der jeweiligen Bahnkonstellation abhängt.

Vom Menü Ansicht aus können Zoomfunktionen (Zoom Uebersicht, Zoom groesser und Zoom kleiner) zur Auswahl des anzuzeigenden Arbeitsbereiches aufgerufen werden. Üblicherweise wird man hierbei mit Zoom Uebersicht die Übersichtsdarstellung wählen, die auch per Default eingestellt ist. Die Funktionen zum Ändern des Zoomfaktors (Zoom groesser, Zoom kleiner) und Funktion Bildneuaufbau können nur dann ausgeführt werden, wenn die Layoutdaten bereits geladen sind.

Mit der Funktion Farbpalette aus dem Menü Ansicht kann die aktuelle Farbzuordnung definiert werden. Mit Farben laden lässt sich eine spezielle Farbtabelle laden. Beim Aufruf des **Autorouters** wird automatisch die Farbtabelle mit dem Namen **`standard`** (aus der Datei **`ged.dat`** im BAE-Programmverzeichnis) geladen. Es empfiehlt sich in jedem Fall, die Farbpalette so einzustellen, dass alle Routinglagen angezeigt werden. Beim Routen mit partiellen Durchkontaktierungen kann es sinnvoll sein, die Farbeinstellung so zu definieren, dass die verwendeten Via-Typen unterschieden werden können.

Die Funktion Breitendarstellung (Default-Einstellung 1.5mm) aus dem Menü Ansicht ermöglicht unter Berücksichtigung des Zoomfaktors die Darstellung der Leiterbahnen in ihrer wahren Breite. Alle Leiterbahnen, deren Breite über dem eingestellten Wert liegen, werden in ihrer wahren Breite relativ zum Zoomfaktor, alle schmäleren als Center-Linie dargestellt.

Die Funktion Potentialanzeige aus dem Menü Ansicht ermöglicht die Darstellung von Verbindungen zu Potentialflächen durch ein Markierungskreuz (Default) oder eine Schwerpunktlinie. Bei der Markierungskreuz-Anzeige werden die Unroutes in Netzen mit Kupferflächen durch einen Punkt (Markierungskreuz) auf den entsprechenden Pins gekennzeichnet. Die andere Variante bewirkt die Freigabe von Vektor-Unroutes zum Schwerpunkt der Kupferfläche; d.h. die Unroutes in Netzen mit Kupferflächen, (jedoch ohne Versorgungslagen) werden mit zwei Punkten zum nächstgelegenen Schwerpunkt einer potentialmäßig geeigneten Kupferfläche (Schwerpunktlinie) dargestellt.

User Language

Im **Autorouter** ist der **Bartels User Language Interpreter** integriert, d.h. vom **Autorouter** aus können **User Language**-Programme gestartet werden. Der Anwender hat damit die Möglichkeit, eigene Zusatzfunktionen nach anwender- bzw. firmenspezifischen Bedürfnissen zu implementieren und in den **Autorouter** einzubinden. Hierzu zählen zum Beispiel Statusanzeigen und Parametereinstellungen, Report- und Testfunktionen, Prüf- und Editierfunktionen, spezielle Plotfunktionen, automatische Platzierungs- und Routingfunktionen, Batch-Prozeduren, usw. usf.

Im **Autorouter** können **User Language**-Programme explizit oder implizit aufgerufen werden. Der explizite Programmaufruf erfolgt über den Menüpunkt Anwenderfunktion im Menü Datei. Nach der Aktivierung dieses Menüpunktes ist auf die Abfrage nach dem Programmnamen der Name des aufzurufenden **User Language**-Programms (z.B. **ulprog**) explizit einzugeben. Die Betätigung einer beliebigen Maustaste oder die Eingabe eines Fragezeichens **?** auf die Abfrage nach dem Programmnamen bewirkt hierbei die Aktivierung eines Popupmenüs mit allen aktuell verfügbaren **User Language**-Programmen.

User Language-Programme können auch implizit über die Tastatur aktiviert werden. Diese Art des Programmaufrufs ist immer dann möglich, wenn nicht gerade eine andere interaktive Eingabe über Tastatur erwartet wird. Die Spezifikation des Programmnamens erfolgt dabei implizit durch Drücken einer Taste. Zulässige Tasten sind dabei die Standardtasten (1, 2, ..., 0, a, b, c, ...; entsprechende Programmnamen sind **ar_1**, **ar_2**, ..., **ar_0**, **ar_a**, **ar_b**, **ar_c**, ...) bzw. die Funktionstasten (F1, F2, ...; entsprechende Programmnamen sind dabei **ar_f1**, **ar_f2**, ...).

Der **Autorouter** ermöglicht den ereignisgesteuerten Aufruf von **User Language**-Programmen. Dabei lösen spezielle Ereignisse bzw. Operationen implizit, d.h. automatisch den Aufruf von **User Language**-Programmen mit definierten Namen aus, sofern diese verfügbar sind. Im Einzelnen sind dies die **User Language**-Programme **AR_ST** beim Starten des **Autorouters**, **AR_LOAD** nach dem Laden eines Elements, **AR_SAVE** vor dem Speichern eines Elements, **AR_TOOL** bei Selektion eines Toolbarelements sowie **AR_ZOOM** bei Änderung des Zoomfaktors. Der Aufruf über die Startupsequenz der Interpreterumgebung eignet sich besonders zur automatischen Voreinstellung von modulspezifischen Parametern sowie zur Tastaturprogrammierung und Menübelegung. Der implizite Aufruf von **User Language**-Programmen nach dem Laden bzw. vor dem Speichern von Elementen ermöglicht die automatische Aktivierung elementspezifischer Bearbeitungsparameter wie z.B. des zuletzt selektierten Zoombereichs oder spezieller Farbeinstellungen. Bei Interaktionen in der Werkzeugliste werden die den selektierten Toolbarelementen zugewiesenen Funktionen ausgelöst. Die Änderung des Zoomfaktors kann dazu benutzt werden, Aktualisierungen in Funktionen zur Verwaltung von Entwurfsansichten auszulösen.

Mit der **Bartels User Language** werden darüber hinaus mächtige Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Beachten Sie bitte, dass über die mit der BAE-Software ausgelieferten **User Language**-Programme eine Vielzahl von Zusatzfunktionen implementiert und transparent in die Benutzeroberfläche des **Autorouters** eingebunden sind.

Eine ausführliche Beschreibung der **Bartels User Language** finden Sie im Bartels User Language Programmierhandbuch (Kapitel 4.2 enthält eine Auflistung aller mit der BAE-Software ausgelieferten **User Language**-Programme).

Neuronales Regelsystem

Im **Bartels AutoEngineer** sind eine Reihe mächtiger Zusatzfunktionen mit Auswirkungen auf die Arbeitsweise des **Autorouters** über das integrierte **Neuronale Regelsystem** implementiert. Kapitel 6.3.2 enthält eine Übersicht über die im Leiterkartenlayoutsystem bereitgestellten Regelsystemanwendungen.

4.5.5 Optionen

Bevor Sie den **Autorouter**-Lauf starten, sollten Sie über das Menü bzw. den Dialog Optionen die Parameter einstellen, mit denen der Router die Leiterkarte entflechten soll. Mit Hilfe dieser Parameter werden prinzipiell die Designregeln für die automatische Leiterkartenentflechtung festgelegt. Aus diesem Grunde ist - sofern nicht anders vermerkt - die Änderung der nachfolgend beschriebenen Options-Parameter jeweils nur *vor* einem Router-Neustart möglich. Die über Optionen eingestellten Parameter werden bei einer Sicherung der Layoutdaten durch den **Autorouter** in der Projektdatei mit abgespeichert und müssen daher bei späteren **Autorouter**-Läufen nicht nochmals eingestellt werden, sofern das Routing mit den zuvor definierten Parametern durchgeführt werden soll.

Routingraster, Standard-Leiterbreite, Standard-Mindestabstand

Sofern der Gridless-Router des **Autorouters** nicht aktiviert ist (siehe unten), arbeitet der **Autorouter** rasterorientiert. Der Anwender kann *vor* dem Start des Autoroutings das Routingraster festlegen. Dies geschieht mit der Funktion Aufloesung im Menü Optionen. Die möglichen Eingaben sind in Tabelle 4-1 aufgelistet.

Tabelle 4-1: Autorouter-Auflösungen

Auflösung	Via-Versatz	Standard Leiterbreite		Standard Mindestabstand	
		[mm]	[mil]	[mm]	[mil]
1/20 Inch std.	-	0.37	14.6	0.29	11.4
1/40 Inch std.	x	0.32	12.6	0.29	11.4
1/50 Inch std.	-	0.25	9.9	0.23	9.1
1/60 Inch std.	x	0.21	8.3	0.19	7.5
1/80 Inch std.	x	0.16	6.3	0.13	5.1
1/100 Inch std.	-	0.13	5.1	0.10	3.9
1/40 Inch no ofs.	-	0.32	2.6	0.29	1.4
1/60 Inch no ofs.	-	0.21	8.3	0.19	7.5
1/80 Inch no ofs.	-	0.16	6.3	0.13	5.1
1/100 Inch w. ofs.	x	0.13	5.1	0.10	3.9

Beim ersten **Autorouter**-Aufruf ist die Auflösung per Default auf 1/40 Zoll Standard eingestellt. Jede Neueinstellung wird mit dem Layout abgespeichert und setzt automatisch auch die Standard-Leiterbahnbreite und den Standard-Mindestabstand auf Defaultwerte (siehe Tabelle 4-1). Diese beiden Parameter lassen sich *nach* der Festlegung der Auflösung noch verändern (Funktionen Standard Leiterbreite und Standard Mindestabstand). Zu beachten ist hierbei jedoch, dass die Summe dieser beiden Werte den Wert für die Auflösung *nicht* überschreiten darf. Soll z.B. ein SMD-Layout mit einer Auflösung von 1/40 Zoll entflochten werden, dann empfiehlt es sich, nach der Definition der Auflösung die Werte für die Standard-Leiterbahnbreite und den Standard-Mindestabstand auf z.B. jeweils 0.2mm einzustellen, damit der **Autorouter** zur Realisierung normaler Leiterbahnen die Pinkanäle zwischen benachbarten Anschlüssen von SO-Gehäusen nutzen kann.

Die Auflösung 1/60 Zoll erlaubt das Verlegen von zwei Leiterbahnen zwischen benachbarten Anschlüssen von DIL-Gehäusen. Um eine saubere Ausmittelung der Leiterbahnen in den entsprechenden Pinkanälen zu erreichen, wird in diesem speziellen Fall intern die gesamte Routingmatrix um 1/120 Zoll verschoben.

Neben den in Tabelle 4-1 aufgeführten Standard-Routingrastern besteht zusätzlich die Möglichkeit, ein beliebiges Routingraster (z.B. metrisch oder für spezielle Pin-Grids) zu wählen. Dies geschieht mit der Option Anderes Raster im Routingraster-Auswahlmenü der Funktion Aufloesung des Menüs Optionen. In diesem Fall werden die Standard-Leiterbahnbreite und der Standard-Mindestabstand auf den halben Wert des gewählten Routingrasters eingestellt (also z.B. auf 0.55mm, wenn ein Raster von 1.1mm gewählt wurde).

Der **Bartels AutoEngineer** verfügt über eine Offgrid-Erkennung. Dadurch kann der **Autorouter** auch Pins anschließen, die nicht im aktuell eingestellten Routingraster liegen (z.B. Stecker mit metrischem Pin-Grid). Grundsätzlich sollte der Anwender jedoch bereits bei der Bauteilplatzierung dafür sorgen, dass - soweit möglich - die Bauteilanschlüsse im Routingraster liegen. Andernfalls kann sich der Rechenzeitbedarf aufgrund der zeitintensiven Offgrid-Erkennung signifikant erhöhen, oder aber bestimmte Anschlüsse können im vorgegebenen Routingraster überhaupt nicht realisiert werden.

Bei der Wahl der Auflösung sollte man darauf achten, dass man kein unnötig feines Routingraster einstellt, da der Speicherplatzbedarf für die Routingmatrix quadratisch mit der Verkleinerung des Rasters anwächst. So wird für eine Routingmatrix im 1/80 Zoll Raster viermal so viel Speicher benötigt wie im 1/40 Zoll Raster. Bei einer Verkleinerung des Routingrasters ist darüber hinaus zu bedenken, dass auch der Rechenzeitbedarf für die Entflechtung dramatisch anwächst, da die Anzahl der vom **Autorouter** zu bewertenden möglichen Wegevarianten in einem noch viel stärkeren Maß als der Speicherplatzbedarf zunimmt.

Bei fast allen Auflösungen kann angegeben werden, ob mit oder ohne Via-Versatz geroutet werden kann. Durch das Versetzen der Vias kann die Anzahl der durch Durchkontaktierungen gesperrten Kanäle drastisch verringert und damit das Routingergebnis entsprechend verbessert werden (siehe Abbildung 4-5). Die Standardwerte sind für Vias kleiner 1mm berechnet. Sofern größere Vias verwendet werden, sollte der Via-Versatz abgeschaltet werden, da sich sonst das Routingergebnis dramatisch verschlechtern könnte.

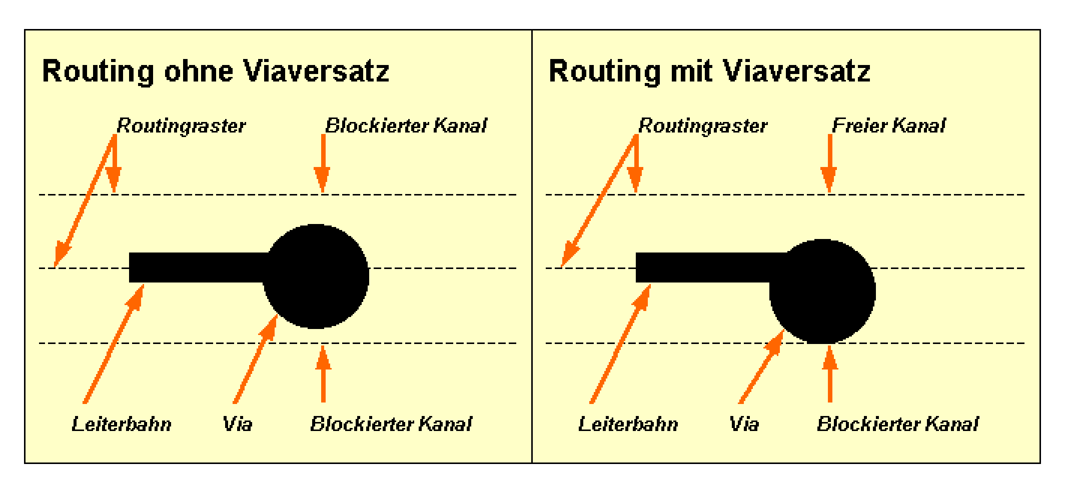


Abbildung 4-5: Routen mit und ohne Via-Versatz

Die Standard-Leiterbreite gibt an, mit welcher Leiterbahnbreite der **Autorouter** die Verbindungen verlegen soll. Ausnahmen hiervon bilden die Signalnetze, für die spezielle Netzattribute definiert sind. Ist für ein Signal das Netzattribut **`routwidth`** mit einer entsprechenden Angabe für die Leiterbahnbreite definiert, dann wird das gesamte Netz in dieser Breite geroutet. Ist für ein Signal das Netzattribut **`powwidth`** definiert, dann werden die in der Bauteilbibliothek vordefinierten Anschlüsse zur Stromversorgung mit der entsprechenden Leiterbahnbreite angeroutet. Der Standard-Mindestabstand gibt an, welche Abstände beim Verlegen der Leiterbahnen von signalfremdem Kupfer mindestens einzuhalten sind. Ausnahmen hiervon bilden die Signalnetze, für die über das Netzattribut **`mindist`** ein spezieller Mindestabstand zu anderen Potentialen vorgegeben ist. Signale, für die Netzattribute definiert sind, werden vom **Autorouter** automatisch mit höherer Priorität bearbeitet. Darüber hinaus kann über das Netzattribut **`priority`** für jedes Signalnetz explizit eine Routingpriorität vorgegeben werden. Weitere Ausführungen zur Definition von netzspezifischen Attributen zur Routersteuerung finden Sie in der Beschreibung des Utilityprogramms **LOGLIB** (siehe Kapitel 7.11 dieses Handbuchs).

Routinglagenzahl und Lagenzuordnung

Über die Funktionen Routinglagenzahl und Lagenzuordnung kann die Anzahl der Signallagen sowie der Typ der Lagen definiert werden. Dabei kann für jede Lage wahlweise eine Vorzugsrichtung (horizontal, vertikal, oder keine) angegeben werden. Es besteht auch die Möglichkeit, Sperrlagen zu definieren, auf welchen der **Autorouter** keine Leiterbahnen verlegen darf. Die Lagenzahl ist per Default entsprechend der im **Layouteditor** vorgenommenen Definition der Obersten Lage (Bestueckseite), die Vorzugsrichtungen sind abwechselnd auf horizontal und vertikal (beginnend bei Signallage 1) eingestellt.

Für die Routinglagenzahl können Werte von 2 bis 12 angegeben werden, d.h. der **Bartels AutoEngineer** ist in der Lage, bis zu 12 Signallagen simultan zu routen. Die Zählweise der Lagen beginnt mit der Signallage 1 (Lötseite). Der **Autorouter** benutzt alle nicht als Sperrlagen definierte Routinglagen für das Verlegen von Leiterbahnen. Die Entflechtung von Single-Layer-Platinen ist damit z.B. mit einer Routinglagenzahl von 2 und der gleichzeitigen Definition der Lage 1 (Lötseite) zur Sperrlage mit Hilfe der Lagenzuordnung möglich.

Alle für die Enflechtung relevanten Objekte auf Sperrlagen werden vom **Autorouter** entsprechend berücksichtigt. Diesen Umstand und die Möglichkeit der Definition von Sperrlagen kann man sich z.B. zunutze machen, um Via-Sperrflächen zu vorzugeben. Hierzu sind z.B. bei einem Zwei-Lagen-Layout zunächst im **Layouteditor** entsprechende Sperrflächen auf der Signallage 3 zu definieren; anschließend kann der **Autorouter** aufgerufen werden, die Routinglagenzahl ist auf 3 einzustellen und die Routinglage 3 als Sperrlage zu definieren. Danach kann das Routing gestartet werden, wobei der **Autorouter** die Signallagen 1 und 2 für die Entflechtung benutzt und zugleich die auf der Signallage 3 definierten Sperrflächen als Bereiche erkennt, in welchen er keine (über alle Signallagen definierten) Durchkontaktierungen setzen darf.

Die Definition der Lagenzuordnung (bei vorgegebener Routinglagenzahl) ist auch zwischen unterschiedlichen Routerläufen (d.h. ohne Router-Neustart) möglich. Dabei ist allerdings zu beachten, dass bei der Definition von Sperrlagen die darauf bereits gerouteten Leiterbahnen in nachfolgenden Routerläufen nur dann umverlegt bzw. wieder entfernt werden, wenn sich dadurch keine Verschlechterung des Routergebnisses ergibt, d.h. wenn dadurch die Anzahl der bereits gerouteten Verbindungen nicht abnimmt.

Wenn z.B. bei einer Multilayer-Platine mit SMD-Bestückung die Anforderung besteht, die Außenlagen zunächst nur zur Ankontaktierung der SMD-Anschlüsse in die Innenlagen zu verwenden, dann könnte man (über Programm-Setup und Programm-Start) zunächst einen Router-Pass zum Vorverlegen von SMD-Vias starten. Anschließend werden die Außenlagen zu Sperrlagen definiert und es wird z.B. ein komplettes Initialrouting - ggf. auch mit anderen Vorzugsrichtungen, als beim Vorverlegen von SMD-Anschlüssen - durchgeführt. Ergebnis wäre ein Layout, dessen Außenlagen nur die SMD-Anschlüsse in die Innenlagen enthalten, während die eigentliche Entflechtung in den Innenlagen vorgenommen wurde.

Der **Bartels AutoEngineer** kann außer den normalen Signallagen noch bis zu 12 Versorgungslagen in den Routingprozess mit einbeziehen. Auf diesen Versorgungslagen können zusätzlich Potentialflächen definiert sein (Split Power Planes). Bei der Verwendung von Versorgungslagen erzeugt der **Autorouter** über Durchkontaktierungen automatisch die korrekten Anschlüsse der SMD-Pins in die Versorgungslagen bzw. zu den darin definierten Potentialen (Power Layer Routing bzw. Split Power Plane Routing). Damit können vom **Bartels AutoEngineer** Multilayer-Platinen mit bis zu 24 Lagen entflochten werden.

Maximal zulässige Viaanzahl

Mit der Funktion Max. Via-Zahl kann die maximal zulässige Anzahl Vias pro Verbindung festgelegt werden. Die Default-Einstellung ist 20. Wird der Wert 0 definiert, dann versucht der Autorouter, das Layout ohne Verwendung von Vias zu routen. Dieser Optionsparameter kann auch zwischen unterschiedlichen Routerläufen (d.h. ohne Router-Neustart) geändert werden. Dabei ist allerdings zu beachten, dass bei einer Reduzierung der maximal zulässigen Viaanzahl nur dann Vias eliminiert werden, wenn sich dadurch keine Verschlechterung des Routergebnisses ergibt, d.h. wenn dadurch die Anzahl der bereits gerouteten Verbindungen nicht abnimmt.

Via-Raster

Mit Via Raster lässt sich das Raster für die Platzierung der Durchkontaktierungen auf 1/10 oder 1/20 Zoll einstellen. Per Default ist hier kein Raster vorgegeben (Rasterfrei). Dieser Optionsparameter kann auch zwischen unterschiedlichen Routerläufen (d.h. ohne Router-Neustart) geändert werden. Dabei ist allerdings zu beachten, dass bei einer Einschränkung des Via-Rasters nur dann Vias umplatziert bzw. eliminiert werden, wenn sich dadurch keine Verschlechterung des Routergebnisses ergibt, d.h. wenn dadurch die Anzahl der bereits gerouteten Verbindungen nicht abnimmt.

Halbraster

Das Menü Optionen enthält die Funktion Halbraster mit den Auswahlpunkten Standard 1:1 und Halbgrid 1:2. Mit Hilfe dieser Option kann gewählt werden, ob das über Aufloesung eingestellte Routingraster im Verhältnis 1:1 eingehalten werden soll (Standard 1:1, Default), oder ob der Router zusätzlich wahlweise auf einem um den halben Rasterabstand verschobenen Grid routen darf (Halbraster 1:2). Erlaubt man dem Router die Entflechtung auf dem Halbraster, dann kann er also z.B. bei einer Standard-Auflösung von 1/40 Zoll - sofern nötig - zusätzlich das Routingraster 1/80 Zoll benutzen, wobei allerdings die aktuell für das Standardraster definierten Werte für die Standard-Leiterbahnbreite und den Standard-Mindestabstand unverändert Gültigkeit besitzen.

Gridless Routing im Neuronalen Autorouter

Im **Autorouter** ist ein regelgesteuerter, rasterfrei arbeitender Router (Gridless Router) integriert. Normalerweise ist der Gridless Router deaktiviert. Dies entspricht der Einstellung Im Raster Routen in der Funktion Gridless des Menüs Optionen. Zur Aktivierung des Gridless Routers ist über die Funktion Gridless einfach die Option Rasterfrei Routen zu selektieren. Der Gridless Router arbeitet *selektiv*, d.h. rasterfreies Routen wird nur lokal an solchen Stellen durchgeführt, wo dies eine Verbesserung hinsichtlich der Entflechtbarkeit bzw. der Fertigungsfreundlichkeit bewirkt. Beim rasterfreien Routen ergeben sich sehr viel mehr Möglichkeiten der Belegung von nicht im Raster liegenden Pinkanälen, wodurch die Entflechtbarkeit komplexer Layouts entscheidend erhöht werden kann. Offgrid-platzierte Pins, d.h. Pins, die nicht im Raster liegen, werden durch den Gridless Router in der Regel auf direkterem Wege angeschlossen. Hierbei kann der Router auch über längere Strecken hinweg vom Raster abweichen, wodurch zum einen die Blockierung benachbarter Pins bzw. Pinkanäle vermieden werden kann, und zum anderen das Layout insgesamt besser für die Fertigung optimiert wird.

Im Modus Rasterfrei Routen kann der Router zudem zwischen rasterfrei platzierten Pins unter Berücksichtigung (d.h., in diesem Fall besser gesagt Ausnutzung) der aktuell eingestellten Mindestabstände hindurchrouten. Damit lassen sich speziell bei dichtbepackten SMD-Platinen erheblich bessere Routingergebnisse erzielen.

Im Gridlessmodus Rast.fr. Pins/Vias werden lediglich die Anschlussegmente an rasterfrei platzierten Pins solange wie möglich gerade gehalten bevor ins Routingraster abgeknickt wird. Beim normalen Routen von Leiterbahnen werden in diesem Modus keine Gridlessabfragen durchgeführt, d.h. es kann z.B. nicht allen Fällen wo dies möglich wäre zwischen rasterfrei platzierten Pins hindurchgeroutet werden.

Warnung

Beachten Sie, dass aufgrund der Verwendung zusätzlicher interner Datenstrukturen (Priority Tree) für das rasterfreie Routen ein gesteigerter Bedarf an Hauptspeicher und Rechenleistung entsteht.

Bahnen Ongrid/Offgrid

Die Funktion Bahnen Ongrid ermöglicht die Festlegung, ob Leiterbahnknicke nur auf dem Routingraster liegen dürfen, oder auch Offgrid (Keine Festlegung, Default-Einstellung) abgeknickt werden darf. Befinden sich auf der Leiterkarte sehr viele rechteckige bzw. quadratische Lötaugen, dann empfiehlt es sich u.U., auf Nur Ongrid Bahnen einzustellen, damit die Mindestabstände zwischen geknickten Leiterbahnsegmenten und Lötaugenecken nicht unterschritten werden (siehe hierzu Abbildung 4-6).

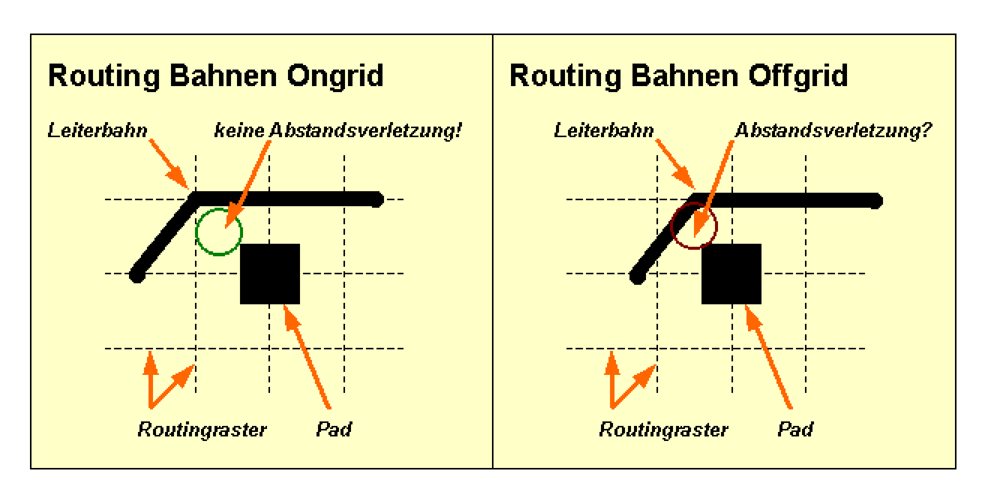


Abbildung 4-6: Routen Bahnen Ongrid/Offgrid

Pin-Anschlussart

Die Funktion Pin Anschluss erlaubt die Definition der Pin-Anschlussart. Hierbei kann wahlweise festgelegt werden, ob die Anschlüsse an rechteckige bzw. annähernd rechteckige Pins über die Pinecken ausgeführt werden dürfen (Ecken freigeben; Default-Einstellung), oder ob diese Anschlussart unterdrückt werden soll (Ecken sperren). Wird das Anrouten an die Pinecken gesperrt, dann ergibt sich i.d.R. ein schöneres (weil gleichmäßigeres) Leiterbild; allerdings kann diese Pin-Anschlussart das Erreichen der 100%-Entflechtung erschweren. Weiterhin ist bei gesperrten Pinecken zu beachten, dass das Anrouten an Pins mit etwa Leiterbreite (oder kleiner) fehlschlagen kann (zu achten ist insbesondere auf breite Leiterbahnen; Workaround evtl.: Vorverlegen und Fixieren). Auch können sich beim Busrouting mit gesperrten Pinecken u.U. unerwünschte Nebeneffekte einstellen.

4.5.6 Steuerung

Über das Menü Steuerung können verschiedene Router-Steuerungsparameter eingestellt werden.

Optimierungszahl

Mit der Funktion Optimierungszahl kann die Anzahl der Optimierungsdurchläufe für die Voll-Autorouter-Funktion festgelegt werden (Wertebereich 0..99, Default 2). Die Optimierungszahl gibt an, wie viele Optimierungen nach dem Erreichen einer 100-prozentigen Entflechtung durch den Voll-Autorouter durchzuführen sind.

Router-Cleanup, Optimierer-Cleanup

Cleanup-Läufe (Anzeige `Pattern Suche`) sind netzübergreifende Via-Optimierungen, die eine gewisse Rechenzeit benötigen, jedoch zu einer drastischen Reduzierung der Viaanzahl führen. Über die Funktionen Router Cleanup bzw. Optimierer Cleanup können diese Cleanup-Läufe im Rip-Up-Router bzw. im Optimierer ein- (Default) oder abgeschaltet werden. Durch die während der Cleanup-Läufe aktive netzübergreifende Mustererkennung kann der **Autorouter** erkennen, welche Leiterbahnverläufe die Realisierung noch nicht verlegter Verbindungen bzw. das Umverlegen bereits gerouteter Bahnen blockieren. Bei komplexen Aufgabenstellungen und insbesondere für das Rip-Up-Routing ist daher von einem Abschalten der Cleanup-Läufe dringend abzuraten. Wenn die Patternsuche deaktiviert ist, dann erfolgt einfach eine sequentielle Abarbeitung der Verbindungen. Dies kann durchaus ausreichend sein für abschließende Optimiererläufe, wobei allerdings zu bedenken ist, dass für die Verschiebung oder Begradigung ganzer Leiterbahnbündel dann ggf. mehrere Durchläufe des Fertigungsoptimierers notwendig sind.

Rip-Up Baumzahl, Rip-Up Tiefe, Rip-Up Retries

Mit den Funktionen Rip-Up Baumzahl (maximale Anzahl der gleichzeitig herausnehmbaren Verbindungen pro Rip-Up-Suchlauf, der Defaultwert hierfür ist 2, der Wertebereich für mögliche Vorgaben erstreckt sich von 1 bis 9), Rip-Up Tiefe (maximale Rip-Up-Baumtiefe, Defaultwert 50, Wertebereich 1 bis 999) und Rip-Up Retries (maximale Anzahl Wiederholungen des Rip-Up-Suchlaufs für eine spezielle Leiterbahn; Defaultwert 2, Wertebereich 0 bis 99) kann die Hartnäckigkeit und Intensität des Rip-Up-Routers beeinflusst werden. Es empfiehlt sich z.B. die Steuerparameter für den Rip-Up-Router auf Maximalwerte zu setzen, wenn nur noch sehr wenige Leiterbahnen zu verlegen sind. Dadurch spart man sich u.U. zeitaufwändige Zwischenläufe des Optimierers während des Rip-Up-Routings, oder aber man erkennt sehr schnell, ob die noch fehlenden Verbindungen mit den aktuellen Vorgaben und Routeroptionen überhaupt realisierbar sind. Der Rip-Up-Router erkennt sehr schnell, wenn z.B. ein Pinanschluss durch Sperrflächen oder vorverlegte und fixierte Leiterbahnen blockiert ist. In diesem Fall wird der Rip-Up-Router nicht endlos lange versuchen, die entsprechende Verbindung zu realisieren; vielmehr wird das Rip-Up-Routing dann abgebrochen, und es erfolgt die Aktivierung der nachgeschalteten Fertigungsoptimierung.

Wenn nach einem Rip-Up-Lauf mehr als 99.5% der Verbindungen gefunden sind, werden die Rip-Up-Parameter automatisch auf Rip-Up Baumzahl 6, Rip-Up Level 200 und Rip-Up Retries 10 gesetzt, sofern nicht bereits höhere Werte eingestellt sind. Damit lassen sich ggf. die langwierigen Optimierer-CLeanupläufe zwischen Rip-Up-Prozessen vermeiden, wenn nur noch wenige Verbindungen offen sind.

SMD Via-Vorlegen

Der **Bartels AutoEngineer** verfügt über ein spezielles Routingverfahren zur Vorverdrahtung von SMD-Anschlüssen (SMD-Fanout-Routing). Dieser Initiallauf kann mit Hilfe der Funktion SMD Via-Vorlegen im Menü Steuerung abgeschaltet (Default; Via Vorlegen Aus) oder aktiviert werden (Via Vorlegen Ein). Ist die Option zur Durchführung der SMD-Via-Vorverdrahtung aktiviert, dann werden - soweit möglich bzw. sinnvoll - in einem dem eigentlichen Initialrouter-Lauf vorgeschalteten speziellen SMD-Via-Routerlauf alle zu verdrahtenden SMD-Pins mit Leiterbahnanschlüssen zu Durchkontaktierungen versehen, wobei sich die Routing-Richtung sinnvollerweise nach Form und Lage der anzuschließenden SMD-Pads ausrichtet. Hierbei bleiben die über die Lagenzuordnung vorgegebenen lagenspezifischen Vorzugsrichtungen unberücksichtigt, um die Blockierung von PLCC/SMD-Pinkanälen bzw. PLCC/SMD-Anschlüssen, die entgegen der Vorzugsrichtung angeordnet sind, zu vermeiden. Durch diese Art der Vorverdrahtung soll beim SMD-Routing mit mehr als zwei Signallagen möglichst frühzeitig eine einigermaßen sinnvolle Vorgabe für die weitgehende Vermeidung der extensiven Nutzung der SMD-Außenlagen erreicht werden. SMD-Pins, die bereits über fixierte Leiterbahnen verbunden sind, werden von diesem Verfahren ausgenommen. Spätestens bei der Optimierung werden eventuell überflüssige SMD-Via-Routes wieder eliminiert. Bei der Verwendung des Verfahrens zur Vorverdrahtung von SMD-Anschlüssen ist allerdings zu beachten, dass u.U. überflüssige Durchkontaktierungen erzeugt werden, und sich der **Autorouter** dadurch relativ frühzeitig die Möglichkeit einer einfacheren Verdrahtung verbaut. Dies kann in speziellen Fällen sowohl zu einer Verschlechterung des Routingergebnisses als auch zu einem dramatischen Anstieg der Rechenzeit führen, da es ggf. für den Router sehr mühselig wird, die frühzeitig blockierten Bereiche durch Rip-Up-Patternsuche wieder zu bereinigen.

Platzierungsoptimierung im Rip-Up-Router

Der Rip-Up/Retry-Router des **Autorouters** bedient sich integrierter Funktionen zur Platzierungsoptimierung durch selektive Bauteil- und Pin/Gate-Swaps. Dieses Verfahren kann über die Funktion Router P/G-Swap im Menü Steuerung wahlweise aktiviert (Option Router Pinswap Ein) bzw. deaktiviert (Option Router P/G-Swap; Default) werden. Ist die Platzierungoptimierung aktiviert, dann führt der Rip-Up-Router nach Bedarf Pin/Gate-Swaps durch, um die Entflechtbarkeit der getauschten Bauteile, Gatter oder Pins zu erhöhen bzw. überhaupt zu ermöglichen und dadurch schneller zu einem komplett gerouteten Layout zu gelangen. Die Auswahl der Swaps erfolgt selektiv, d.h. es werden nur solche Swaps ausgeführt, die auch tatsächlich eine Vereinfachung des Entflechtungsproblems bewirken. Die Platzierungsoptimierung unterliegt natürlich dem Backtrackingverfahren des Routers, d.h. Swaps, die eine Verschlechterung des aktuellen Routingergebnisses bewirken, werden automatisch zurückgenommen. Selbstverständlich wird die Zulässigkeit der Swaps anhand der jeweiligen Bauteildefinitionen aus der Logischen Bibliothek überprüft. Von der Platzierungsoptimierung ausgenommen bleiben fixierte Bauteile sowie Pins, die bereits über fixierte Leiterbahnen angeschlossen sind.

Wurden vom **Autorouter** Pin/Gate-Swaps durchgeführt, dann wird beim Speichern des Layouts ein spezieller Datenbankeintrag erzeugt, der im **Schaltplaneditor** ausgewertet wird und die automatische Durchführung notwendiger Backannotation-Prozesse beim Laden von Schaltplänen aktiviert (siehe hierzu Kapitel 2.7. Im **Packager** wird dieser Datenbankeintrag ebenfalls ausgewertet, um den Anwender ggf. über eine Bestätigungsabfrage auf die Notwendigkeit zur Durchführung der Backannotation vor dem nächsten **Packager**-Lauf hinzuweisen (siehe hierzu Kapitel 3.2.3).

Zwischenspeichern

Die Funktion Zwischenspeichern ermöglicht das automatische Zwischenspeichern von Teil-Routingergebnissen (Default) oder die Abschaltung dieser Funktion.

4.5.7 Strategie

Über das Menü Strategie können verschiedene Router-Strategieparameter eingestellt werden. Die hierbei änderbaren Kostenfaktoren sollten nur in Ausnahmefällen modifiziert und erst recht nicht auf Extremwerte gesetzt werden, da die Default-Einstellungen Erfahrungswerte sind, die für den Großteil der Routingprobleme die bestmögliche Routerstrategie darstellen.

Bei einer Änderung der Strategieparameter ist insbesondere zu bedenken, dass sich diese Parameter in den meisten Fällen in hohem Maß wechselseitig beeinflussen. So hat z.B. die Elimination von Vias (z.B. durch hohe Via-Kosten) in aller Regel zur Folge, dass in höherem Maße gegen die Vorzugsrichtungen verstoßen werden muss. Dadurch wird die Einstellung eines hohen Kostenfaktors für die Einhaltung der Vorzugsrichtung möglicherweise völlig kompensiert wenn nicht gar bedeutungslos.

Des Weiteren ist zu beachten, dass mit den Strategieparametern lediglich untergeordnete Optionen für die Entflechtung vorgegeben werden können. Viel wichtiger als z.B. die Einhaltung von Vorzugsrichtungen ist für den **Autorouter** die komplette Lösung des gestellten Verdrahtungsproblems. Der **Autorouter** versucht in jedem Fall die Leiterkarte zu 100% zu entflechten, wobei bestimmte Strategieparameter ggf. zunächst völlig unberücksichtigt bleiben müssen.

Tabelle 4-2: Autorouter-Strategieparameter

Strategieparameter	Wertebereich	Default	Wirkung im Router	Wirkung im Optimierer
V-Richtung optim.	Normal Vorzugsrichtung Diagonal	Normal	-	x
Via-Kosten	2..20	10	x	x
Pinkanal-Kosten	0..10	3	x	-
V-Richtung-Kosten	0..5	1	x	x
R-Aend.-Kosten	0..5	2	-	x
Pack-Kosten	0..5	1	x	-
Stat. Verteilung	0..50	10	x	-
Bus-Abknickkosten	0..5	2	x	-
Abstand-1-Kosten	0..10	5	x	-
Abstand-2-Kosten	0..10	2	x	-
Kreuzungskosten	2..100	20	x	x
Diagonal-Kosten	0..10	5	-	x
Off-Grid-Kosten	0..5	2	x	x
Vorzugsraster	0..7	0	x	x
Anti-Vorzugsraster-Kosten	0..10	1	x	x
Anti-Netzbereich-Kosten (**BAE HighEnd**)	0..5	1	x	x

Vorzugsrichtungs-Optimierung

Über die Funktion V-Richtung optim. kann der Optimierer angewiesen werden, ohne Vorzugsrichtung (Normal, Default), mit Vorzugsrichtung (Vorzugsrichtung, für Schwallbadlötung ohne Lötstoplack) oder vorzugsweise diagonal (Diagonal, 45-Grad) zu optimieren.

Viakosten

Mit Via-Kosten wird der Router angewiesen, mit möglichst wenigen Vias (hoher Wert, möglicherweise kompliziertere Leiterbahnführung) oder unter Verwendung einer größeren Anzahl von Vias (niedriger Wert, u.U. einfachere Leiterbahnführung) zu entflechten.

Pinkanal-Kosten

Mit der Einstellung der Pinkanal-Kosten kann eine Vermeidung der Benutzung oder Belegung von Pinkanälen (hoher Wert) oder eine häufige Belegung der Pinkanäle (niedriger Wert) veranlasst werden. Pinkanäle sind die Bereiche zwischen benachbarten Bauteilanschlüssen.

Vorzugsrichtungskosten

Über V-Richtung-Kosten können die Kosten für das Routen gegen die Vorzugsrichtung eingestellt werden. Ein hoher Wert bedeutet dabei die kontinuierliche Einhaltung der Vorzugsrichtung, während ein niedriger Wert die häufige Verletzung der Vorzugsrichtung zulässt. Die Vorzugsrichtung kann für jede Routinglage mit der Funktion Lagenzuordnung (Menü Optionen) für jede Routinglage entweder auf horizontal oder auf vertikal gesetzt werden. Dieser Kostenfaktor wird beim Routen immer berücksichtigt. Beim Optimieren hingegen werden die Vorzugsrichtungskosten nur dann berücksichtigt, wenn der Optimierer mit der Funktion V-Richtung optim. (siehe oben) angewiesen wurde, die eingestellten Vorzugsrichtungen einzuhalten.

Richtungsänderungskosten

Mit R-Aend.-Kosten werden dem Optimierer die Richtungsänderungskosten vorgegeben. Ein hoher Wert führt zu einer lokalen Vermeidung von Richtungsänderungen, ein niedriger Wert erlaubt eine höhere Anzahl von Leiterbahnknicken.

Packungskosten

Über Pack-Kosten werden die Packungskosten für das Routing vorgegeben. Ein hoher Wert führt zu einer starken Bündelung der Leiterbahnen, ein niedriger Wert erlaubt eine weitere Verteilung der Leiterbahnen (möglicherweise jedoch auf Kosten einer 100%-Auflösung).

Kostenbasis für die Statistische Verteilung

Mit Stat. Verteilung wird die Kostenbasis für die statistische Verteilung der Leiterbahnen vorgegeben. Je höher der Wert, umso höher der Einfluss auf das Routing, d.h. umso gleichmäßiger die globale Verteilung der Leiterbahnen über die Leiterkarte.

Bus-Abknickkosten

Die Bus-Abknickkosten regeln das bei Busverbindungen notwendige Abknicken direkt nach einem durchlaufenen Pinkanal. Ein hoher Wert bewirkt eine hohe Priorität des Abknickens.

Rip-Up-Abstandskosten

Die Abstand-1-Kosten regeln während des Rip-Up-Routings die Vermeidung der Benutzung von Wegen gelöschter Leiterbahnen im Nahbereich. Eine hohe Zahl steht für eine weitgehende Vermeidung.

Die Abstand-2-Kosten regeln während des Rip-Up-Routings die Vermeidung der Benutzung von Wegen gelöschter Leiterbahnen im Fernbereich. Eine hohe Zahl steht für eine weitgehende Vermeidung.

Kreuzungskosten

Die durch Kreuzungskosten einstellbaren Leiterbahn-Übergangskosten steuern die Patternerkennung während der netzübergreifenden Optimierung und haben damit Einfluss auf die Erkennung unnötiger Vias. Ein hoher Wert erlaubt mehr komplexe und viahaltige Leiterbahnkonstruktionen. Ein niedriger Wert führt zur vermehrten Analyse und zeitaufwändigeren netzübergreifenden Optimierung.

Diagonalrouting-Kosten

Die Diagonal-Kosten erlauben (hoher Wert) bzw. verbieten (niedriger Wert) dem Optimierer die vorzugsweise diagonale (45-Grad) Leiterbahnführung. Dieser Kostenfaktor wird nur dann berücksichtigt, wenn der Optimierer mit der Funktion V-Richtung optim. angewiesen wurde, in Diagonal-Richtung zu optimieren.

Offgrid-Routing-Kosten

Mit Off-Grid-Kosten wird für den Fall, dass im Halbraster (siehe oben) geroutet wird, die Priorität der Vermeidung (hoher Wert) oder Benutzung (niedriger Wert) des Halbrasters während dem Routen und Optimieren bestimmt.

Vorzugsraster, Anti-Vorzugsraster-Kosten

Zur besseren Leiterbahnverteilung auf Leiterkarten kann ein Vorzugsraster im Bereich von 0 bis 7 angegeben werden. Mit der Voreinstellung 0 ist kein Vorzugsraster selektiert. Mit höheren Vorzugsrasterwerten wird bevorzugt jeder 2. (bzw. 4., 8., 16., 32., 64. oder 128.) Rasterpunkt geroutet. Mit den Anti-Vorzugsraster-Kosten wird vorgegeben, wie stark ein Routen außerhalb des Vorzugsrasters bestraft werden soll. Vorzugsrastereinstellungen sind nur für Layouts sinnvoll, auf denen entsprechend viel Freiraum vorhanden ist. Die Kostenfaktoren sollten schon beim ersten Start des Autorouters entsprechend eingestellt werden. Ein nachträgliches Optimieren erfordert mehrere Optimiererläufe und kann daher sehr zeitaufwändig werden. Es sei auch darauf hingewiesen, dass eine entsprechend hohe Einstellung der Kostenfaktoren ein Routen von Umwegen begünstigen kann.

Anti-Netzbereich-Kosten (BAE HighEnd)

Der Strategieparameter Anti-Netzbereich-Kosten wird nur in **BAE HighEnd** unterstützt. Damit wird vorgegeben, wie stark ein Routen außerhalb vorgegebener netzspezifischer Routingbereiche bestraft werden soll. Ein hoher Wert weist den Router an, netzspezifische Routingbereiche nach Möglichkeit nicht zu verlassen, ein niedriger Wert gestattet ein häufigeres Verlassen vorgegener netzspezifischer Routingbereiche. Für diesen Strategieparameter können Werte von 0 bis 5 angegeben werden, die Standardeinstellung ist 1.

4.5.8 Autorouterprozeduren

Der **Autorouter** unterstützt verschiedene Autorouting-Algorithmen bzw. **Autorouter**-Prozeduren wie Voll-Autorouter, Initialrouting, SMD-Fanout-Routing, Rip-Up/Retry-Routing, Optimierer, Änderungsrouten, usw. Die entsprechenden Funktionen zur Aktivierung dieser Routingprozesse finden sich in den Menüs Autorouter (Voll-Autorouter, Optimierer, Einlesen Bahnen, Programm-Setup und Programm-Start), Steuerung (SMD-Via-Vorlegen) und Interaktion (Single Net Route, Bauteil Routen Platzieren/routen). Wird im **Neuronalen Autorouter** eine dieser Prozeduren aktiviert, *nachdem* über das Menü Optionen grundlegende Routerparameter (Lagenanzahl, Routingraster, Freigabe/Sperren des Halbrasters, Standard-Leiterbreite, Standard-Mindestabstand oder Pinanschluss-Verfahren) geändert wurden, dann wird automatisch das aktuelle (unfixierte) Routingergebnis verworfen und die selektierte **Autorouter**-Prozedur wird mit den geänderten Optionen neu aufgesetzt.

Beachten Sie, dass die Autourouting-Prozesse in **BAE HighEnd** aufgrund der Verwendung effizienterer, interner Datenstrukturen (HighSpeed Kernel) um etwa 30 Prozent schneller ablaufen als in **BAE Professional**.

Nach dem Start einer **Autorouter**-Prozedur kann der Routvorgang am Bildschirm verfolgt und jederzeit durch Betätigung einer beliebigen Taste angehalten bzw. abgebrochen werden. Der Abbruch des Routvorgangs bewirkt automatisch auch eine Sicherung des aktuell besten Routergebnisses.

Initialrouter Einzeldurchgang

Diese Prozedur führt einen einzelnen Durchgang des Initialrouters mit einer vorgegebenen Kanalbreite und einer maximal zulässigen Anzahl von Durchkontaktierungen pro Verbindung durch. Die Routingkanalbreite ist dabei in Rasterpunkten zu spezifizieren und gibt die einseitige maximal zulässige Abweichung von der Vorzugsrichtung an; der Wert 0 für die Kanalbreite entspricht hierbei einer vollständigen Freigabe. Die tatsächliche maximal zulässige Viazahl wird bestimmt durch das Minimum aus dem über den entsprechenden Optionsparameter mit der Funktion Max. Via-Zahl eingestellten Wert und dem für den Initialrouting-Lauf angegebenen Wert.

Initialrouter Komplettdurchgang

Diese Prozedur aktiviert automatisch alle Standarddurchgänge des Initialrouters, um alle ohne Rip-Up vom Router realisierbaren offenen Verbindungen zu verlegen. Es werden hierbei nacheinander automatisch vier Initialrouting-Läufe aktiviert, wobei sukzessive die Kanalbreite und die maximal zulässige Viaanzahl erhöht werden. Der vierte und letzte Initialrouting-Durchlauf wird mit einer Kanalbreite von 0 (keine Einschränkung hinsichtlich der Abweichung von der Vorzugsrichtung) und der über den entsprechenden Optionsparameter eingestellten maximal zulässigen Anzahl an Vias pro Verbindung (siehe oben) durchgeführt.

Rip-Up/Retry-Router

Diese Prozedur aktiviert den Rip-Up/Retry-Router, um noch verbleibende offene Verbindungen zu verlegen. Verbindungen, bei denen kein Rip-Up notwendig ist, werden nach kurzem Test ohne dieses realisiert. Existieren nach einem Rip-Up-Durchlauf noch offene Verbindungen, die ohne Verletzung der Designregeln realisierbar wären, dann wird nach einem zweimaligen Optimiererlauf erneut der Rip-Up-Router gestartet.

Optimierer

Die Funktion Optimierer startet einen einzelnen Durchlauf des Fertigungsoptimierers. Der Optimierer eliminiert unnötige Vias eliminiert, glättet Leiterbahnen und verlegt letztere wenn möglich auch optimaler. Gleichzeitig versucht der Optimierer, die noch fehlenden Verbindungen zu verlegen. Dies geschieht mit einer Kanalbreite von 0 (keine Einschränkung hinsichtlich der Abweichung von der Vorzugsrichtung) und der mit der Funktion Max. Via-Zahl im Menü Optionen eingestellten maximal zulässigen Anzahl von Vias pro Verbindung.

SMD Via-Vorlegen

Die Funktion SMD Via-Vorlegen dient der Vorplatzierung von Vias für alle angeschlossenen SMD-Pins ("SMD Fanout Routing"). Dabei bestimmt die Kanalbreite die maximal zulässige Abweichung von der Vorzugsrichtung. Diese Prozedur wird vom Voll-Autorouter (siehe unten) automatisch als erster Routing-Pass aktiviert, wenn der entsprechende Parameter zur Routersteuerung mit der Funktion SMD Via-Vorlegen im Menü Steuerung (siehe Kapitel 4.5.6) vorher gesetzt wurde.

Voll-Autorouter

Die Funktion Voll-Autorouter führt einen vollautomatischen Router-Durchlauf mit allen Router-Passes (SMD-Vias Vorverlegen, kompletter Initialrouting-Lauf, Rip-Up-Lauf, Optimierung) durch. Dies ist die Standardprozedur für die vollständige

Entflechtung einer Platine. Die Anzahl der Optimierungsläufe ist als Steuerparameter mit der Funktion Optimierungszahl im Menü Steuerung einstellbar. Der Initiallauf zur Vorverdrahtung von SMD-Anschlüssen wird nur dann durchgeführt, wenn der entsprechende Parameter zur Routersteuerung mit der Funktion SMD Via-Vorlegen im Menü Steuerung (siehe Kapitel 4.5.6) vorher gesetzt wurde.

Programm-Setup und Programm-Start

Mit der Funktion Programm-Start kann ein zuvor mit der Funktion Programm-Setup festgelegter Router-Lauf mit bis zu 10 Router-Passes gestartet werden. Dabei können folgende Routingprozeduren aktiviert werden:

Kommando	Routingprozedur
E	Einlesen Bahnen
V	Voll-Autorouter
I	Initialrouter Einzeldurchgang
C	Initialrouter Komplettdurchgang
R	Rip-Up/Retry-Router
O	Optimierer
S	SMD-Via-Vorverlegen
-	keine Funktion (Router-Pass aus Batch entfernen)

Bei Initialrouter Einzeldurchgang sind die Routingkanalbreite und die maximal zulässige Viaanzahl pro Verbindung anzugeben. SMD-Via-Vorverlegen erfordert die Spezifikation der Routingkanalbreite. Optimierer verlangt die Angabe über die Anzahl der durchzuführenden Optimierer-Läufe wobei bis zu 999 Durchgänge aktiviert werden können.

Einzelnetzrouting

Der im **Autorouter** integrierte Single-Net-Autorouter dient der automatischen Entflechtung einzelner, selektierbarer Signalnetze bzw. Verbindungen. Damit können z.B. Stromversorgungsnetze oder kritische Leiterbahnen unter Berücksichtigung spezifischer Vorgaben für Leiterbreiten, Mindestabstände, Lagenzuordnungen, usw. selektiv vorverlegt werden.

Der Single-Net-Router wird mit der Funktion Single Net Route im Menü Interaktion des **Autorouters** aktiviert. Die Auswahl des zu verdrahtenden Signalnetzes erfolgt durch Selektion eines entsprechenden Pins oder einer (vorverlegten) Leiterbahn des gewünschten Netzes.

Component-Routing

Mit der Funktion Bauteil routen aus dem Menü Interaktion können mausselektierte Bauteile verdrahtet werden. Es werden dabei alle am jeweils selektierten Bauteil angeschlossenen Netze geroutet.

Vollautomatische Platzierung und Entflechtung

Mit der Funktion Platzieren/routen aus dem Menü Autorouter werden die nacheinander die Prozeduren Voll-Autoplace und Voll-Autorouter aktiviert. Damit ist eine vollautomatische Platzierung und Entflechtung des Layouts möglich.

Selektive Airlineanzeige und Netzgruppen-Routing

Bei der Auswahl der Mincon-Funktion stehen wie im **Layouteditor** Funktionen zum Ein- und Ausblenden von Netzen für die Airlineanzeige zur Verfügung. Ausgeblendete Netze werden vom **Autorouter** nicht geroutet. Diese Einstellungen können zwischen einzelnen Routerläufen geändert werden. So ist es z.B. möglich, mit Einlesen Bahnen zunächst selektiv einzelne Netze einzulesen und dann die Netzsichtbarkeit anderer Netze zu ändern, um diese neu hinzuzurouten.

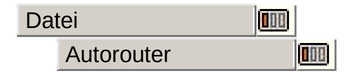
Bereichs- und Blockrouting im Autorouter der BAE HighEnd Version

In **BAE HighEnd** ist im **Autorouter** die Vorgabe von Routingbereichen für Netztypen möglich. Die Routingbereiche sind im **Layouteditor** als Dokumentarflächen auf Signallagen zu definieren. Zusätzlich sind an diese Dokumentarflächen (mit dem **User Language**-Programm **GEDRULE**) Regeln zuzuweisen, aus deren Prädikat `net_type` der Netztyp hervorgeht. Beispielregeln hierfür finden sich in der Datei `nettype.rul`, die bei der Installation im **User Language**-Verzeichnis (`baeulc`) abgelegt wird. Pro Layout können maximal 8 Netztypen für Routingbereiche verwendet werden. In den netztypspezifischen Routingbereichen darf der **Autorouter** dann nur Netze routen, an die über das Netzattribut `$nettype` ein dem Routingbereich entsprechender Netztyp zugewiesen ist. Verbindungen ohne Netztypzuweisung oder mit anderen Netztypzuweisungen sind in den Routingbereichen lediglich zum Ankontaktieren von Pins zulässig. Die Netze eines Netztyps werden bevorzugt in dem hierfür definierten Routingbereich geroutet. Zur Steuerung des Verhaltens außerhalb dieser Bereiche kann über das Menü Strategie der Kostenfaktor Anti-Bereichskosten (0..5) modifiziert werden.

4.5.9 Durchführung des Autoroutings

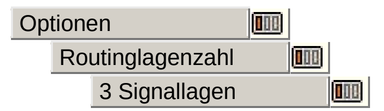
Aufruf des Autorouters

Sofern Sie sich nicht im **Layouteditor** befinden, sollten Sie diesen zunächst aufrufen und das Layout `board` aus der Datei `demo.ddb` laden. Der Aufruf des **Autorouters** erfolgt vom **Layouteditor** aus über das Menü Datei:

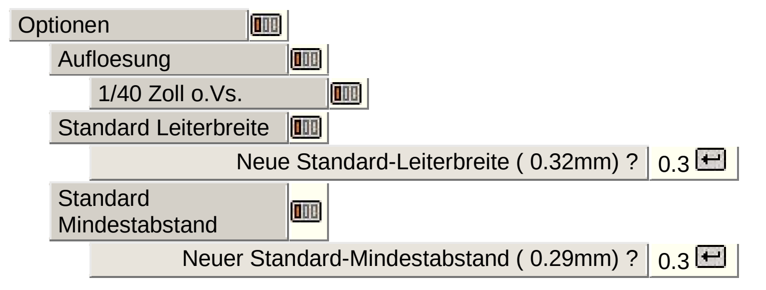


Optionsvorgaben

Es soll nun für das Beispiellayout ein Routing auf 3 Signallagen mit speziellen Router-Passes durchgeführt werden. Stellen Sie zunächst mit den folgenden Kommandos die Routinglagenzahl auf 3 ein:



Stellen Sie mit den folgenden Kommandos die Auflösung auf 1/40 Zoll ohne Via-Versatz, sowie die Standard-Leiterbahnbreite und den Standard-Mindestabstand auf 0.3mm ein:



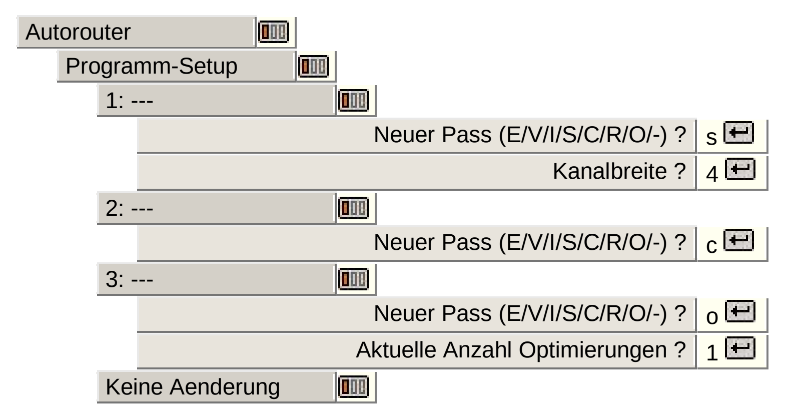
Bilddarstellung, Layoutanzeige

Stellen Sie nun über das Menü Ansicht die Breitendarstellung auf den Wert 0.1mm ein, und rufen Sie die Funktion Zoom Uebersicht auf:

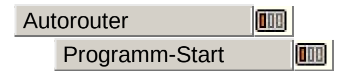


Router-Batch-Lauf

Definieren Sie nun mit der Funktion Programm-Setup einen Router-Batch-Lauf bestehend aus SMD Via-Vorlegen mit Routingkanalbreite 4, Initialrouting Komplettdurchgang sowie einem Optimierer-Lauf:



Die Eingabe eines Bindestrichs auf die Abfrage nach dem neuen Pass bewirkt das Löschen des selektierten Eintrags aus dem Programm-Setup. Die Reihenfolge, in der die Router-Passes mit der Funktion Programm-Setup definiert werden, gibt gleichzeitig die Abarbeitungsreihenfolge an, in der diese Router-Passes anschließend durch die Funktion Programm-Start ausgeführt werden. Starten Sie nun mit dem folgenden Kommando die zuvor mit Programm-Setup definierten Router-Passes:



Das Router-Batchprogramm kann wahlweise auch ohne Umweg über die Beendigung von Programm-Setup und die Aktivierung Programm-Start einfach durch Betätigung des Start-Buttons im Dialog der Funktion Programm-Setup gestartet werden.

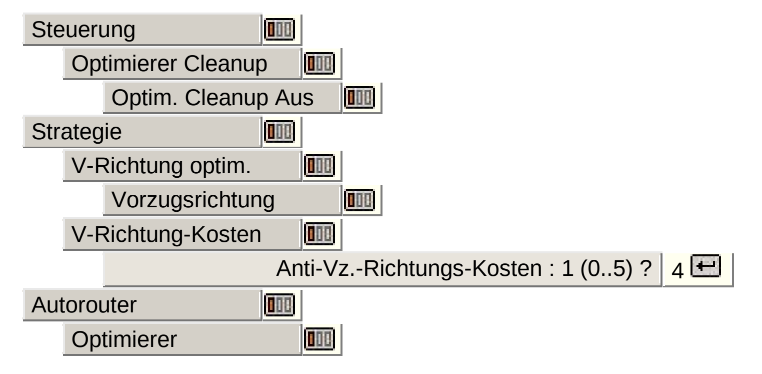
Der erste Router-Pass führt eine Vorverdrahtung der SMD-Anschlüsse durch. Danach wird ein kompletter Initialrouting-Lauf durchgeführt. Nach dem Initialrouting ist das Layout bereits zu 100% entflochten. Abschließend erfolgt noch ein Optimiererlauf. Nach jedem einzelnen Router-Pass wird automatisch das aktuelle Routingergebnis gesichert. Nach Beendigung des letzten Routerlaufes sollte folgende Meldung in der Mitteilungszeile stehen:

```
Max. 53 von 53 Routes (Pins: 76, Vias: 5)
```

Der **Autorouter** hat das Layout also nun fertig entflochten und dabei 5 Durchkontaktierungen gesetzt.

Optimierung

Führen Sie nun mit den folgenden Kommandos einen zusätzlichen Optimiererlauf (ohne Patternerkennung, jedoch mit hoher Priorität zur Einhaltung der Vorzugsrichtung) durch:



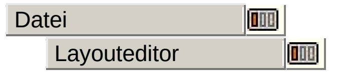
Nach Beendigung des obigen Optimiererlaufes sind die Leiterbahnen soweit möglich entsprechend der eingestellten Vorzugsrichtungen verlegt. Der Preis hierfür ist eine Erhöhung der Viaanzahl auf 13.

Verlassen des Autorouters

Der **Autorouter** kann mit einer der Funktionen Programmende, Hauptmenue oder Layouteditor aus dem Menü Datei verlassen werden.

Die Funktion Programmende bewirkt einen Rücksprung auf die Betriebssystemebene. Die Funktion Hauptmenue bewirkt einen Rücksprung in die Shell des **Bartels AutoEngineer** (mit automatischer Sicherung des bearbeiteten Layouts). Mit der Funktion Layouteditor gelangen Sie (nach automatischer Sicherung des bearbeiteten Layouts) zurück in den **Layouteditor**, wo automatisch das gerade bearbeitete Layout geladen wird.

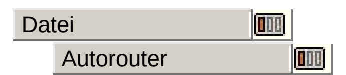
Springen Sie also nun wieder zurück in den **Layouteditor**:



Änderungsrouting

Die Verwendung von drei Signallagen erscheint für unser einfaches Beispiellayout etwas verschwenderisch. Wir wollen daher jetzt in einem erneuten Router-Lauf die Routinglagenzahl auf 2 reduzieren. Die vom vorhergehenden Routerlauf erzeugten Daten sollen dabei mit Hilfe der Funktion Einlesen Bahnen soweit möglich übernommen werden.

Wechseln Sie also wieder in den **Autorouter**:

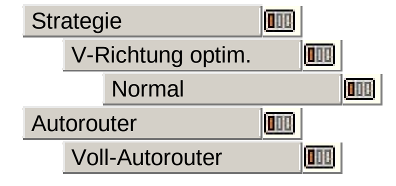


Setzen Sie nun die Routinglagenzahl auf zwei und aktivieren Sie die Funktion Einlesen Bahnen:

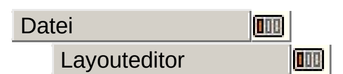


Die Funktion Einlesen Bahnen übernimmt die Routingdaten aus dem vorhergehenden Routerlauf. Da jedoch zwischenzeitlich die Routinglagenzahl auf 2 verringert wurde, können die zuvor auf der Signallage 3 gerouteten Verbindungen nicht wieder geladen werden. Die Anzeige in der Mitteilungszeile meldet daher nach der Ausführung der Funktion Einlesen Bahnen 9 offene Verbindungen (44 von 53 Routes) und eine Viaanzahl von 8 (anstatt 13).

Stellen Sie nun die Vorzugsrichtung für den Optimierer wieder auf Normal und starten Sie den Voll-Autorouter, um die noch fehlenden Verbindungen einzulegen:



Nach dem Voll-Autorouter-Lauf ist das Layout - nun mit einer Viaanzahl von 6 und auf zwei anstatt 3 Signallagen - wieder zu 100% geroutet. Damit ist das Autorouting beendet, und Sie können mit den folgenden Kommandos wieder zum **Layouteditor** zurückspringen.



Das Beispiellayout sollte jetzt entsprechend Abbildung 4-7 aussehen.

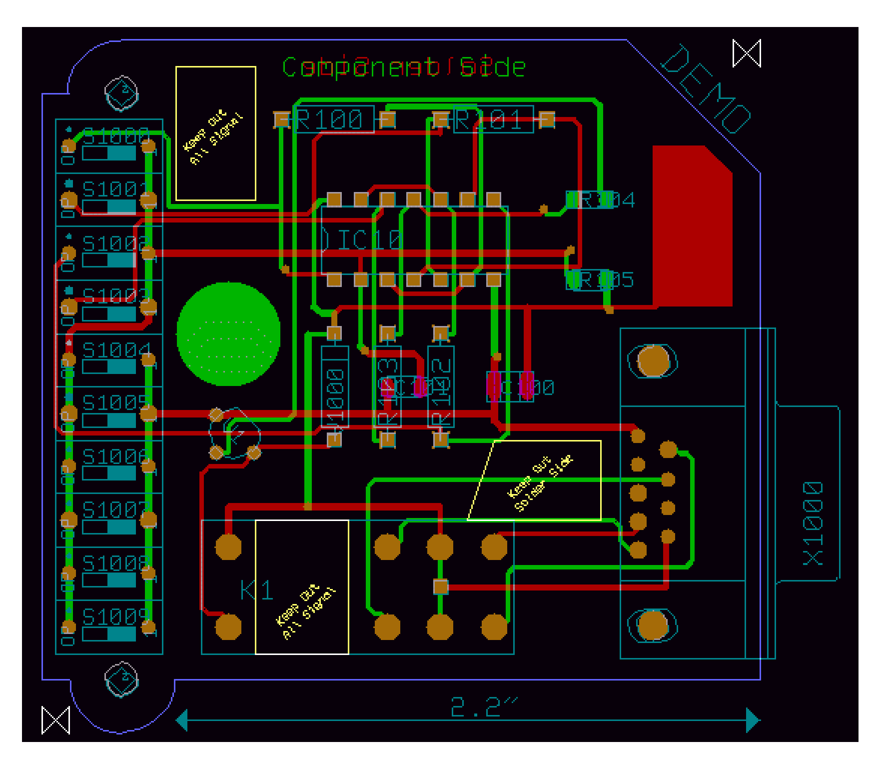


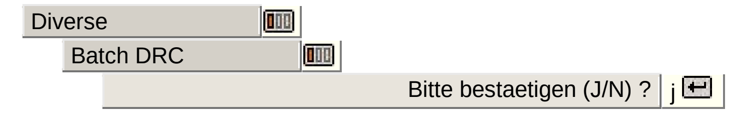
Abbildung 4-7: Automatisch entflochtenes Layout

4.6 Spezielle Layoutfunktionen

Eine Reihe wichtiger Funktionen, die sich zum großen Teil aus der Flexibilität und Konsistenz der Datenstrukturen bzw. des Datenbanksystems ergeben, wurde bisher noch nicht besprochen. Anhand einiger spezieller Funktionen soll in diesem Abschnitt die außerordentliche Vielseitigkeit des **Bartels AutoEngineer** und die enorme Mächtigkeit seiner Funktionen aufgezeigt werden.

4.6.1 Batch-Design Rule Check, Report

Mit der Funktion Batch DRC im Menü Diverse wird ein kompletter Design Rule Check durchgeführt. Wir empfehlen dringendst, diese Funktion vor der Generierung der Fertigungsdaten aufzurufen, um zu prüfen, ob alle Design-Regeln eingehalten wurden, keine Kurzschlüsse auf der Leiterkarte vorhanden sind, und alle Netze geroutet sind. Führen Sie mit den folgenden Kommandos einen Batch Design Rule Check für das im Speicher befindliche Layout **board** aus der Datei **demo.ddb** durch:



Die Funktion Batch DRC aktiviert nach Durchführung der Designprüfung implizit die Funktion Report aus dem Menü Diverse, um das Ergebnis der Prüfung bzw. den Designstatus anzuzeigen. Mit den folgenden Kommandos lässt sich die Funktion Report auch explizit aufrufen:



Das System sollte nun folgende Bildschirmausgabe erzeugen (Null-Fehleranzeigen werden ggf. unterdrückt):

```
Datei : demo.ddb
Typ : Layout / Element : board

Anzahl Netze ....................: 22
Anzahl offene Verbindungen .....: 0
Anzahl Kurzschluesse ...........: 0
Anzahl Kupferabstandsfehler ....: 0
Anzahl Dokumentarabstandsfehler : 0
Anzahl Versorgungsfehler .......: 0
Anzahl fehlender Bauteile ......: 0
Anzahl Bauteiltypfehler ........: 0
Anzahl fehlender Netzlistenpins : 0
Anzahl Hoehenfehler ............: 0
Benutzte Versorgungslagen ......: -
Benutzte Signallagen ...........: 1-2
```

Der Eintrag **Anzahl Netze** gibt die Anzahl der in der Netzliste des aktuell geladenen Layouts definierten Netze an. Der Eintrag **Anzahl offene Verbindungen** gibt die Anzahl der auf dem aktuell geladenen Layout noch nicht gerouteten (Zweipunkt-)Verbindungen an. Der Eintrag **Anzahl Kurzschluesse** gibt die Anzahl der vom Design Rule Check auf dem aktuell geladenen Layout erkannten Kurzschluesse an. Der Eintrag **Anzahl Kupferabstandsfehler** gibt die Anzahl der vom Design Rule Check auf dem aktuell geladenen Element (online) erkannten Mindestabstandsverletzungen auf Kupferlagen an. Der Eintrag **Anzahl Dokumentarabstandsfehler** gibt die Anzahl der vom Design Rule Check auf dem aktuell geladenen Element (online) erkannten Mindestabstandsverletzungen auf Dokumentarlagen an. Der Eintrag **Anzahl Versorgungsfehler** gibt die Anzahl der sich wechselseitig überschneidenden Potentialflächen auf geteilten Potentiallagen (Split Power Planes) an. Der Eintrag **Anzahl fehlender Bauteile** zeigt die Anzahl der auf dem aktuell Layout noch nicht platzierten Netzlisten-Bauteile an. Der Eintrag **Anzahl Bauteiltypfehler** zeigt die Anzahl der Netzlistenbauteile an, die mit einer falschen Gehäusebauform auf dem aktuellen geladenen Layout platziert sind. Der Eintrag **Anzahl fehlender Netzlistenpins** gibt die Anzahl nicht platzierter bzw. fehlender Netzlistenpins an. Der Umstand, dass bei fehlenden Netzlistenpins die Anzahl der offenen Verbindung u.U. nicht korrekt angezeigt werden kann, wird ggf. durch einen entsprechenden Hinweis bei der Anzahl der offenen Verbindungen angezeigt. **Anzahl Hoehenfehler** zeigt die Anzahl der Verstösse gegen eventuell über das Regelsystem definierte (Bauteil-)Höhenrestriktionen an. Die Einträge **Benutzte Versorgungslagen** und **Benutzte Signallagen** dienen der Anzeige der benutzten Versorgungs- und Signallagen. Die Lagen von Lage 1 bis zur obersten Lage gelten immer als benutzt. Mit diesen Einträgen ist insbesondere zu erkennen, ob auf nicht mit einem globalen Netz versehenen Versorgungslagen dennoch für die Connectivity relevante Split-Powerplaneflächen vorhanden sind.

DRC-Fehleranzeigemodus und DRC-Fehlerlokalisierung

Die über die Funktion Einstellungen aus dem Menü Ansicht aktivierbare Dialogbox enthält den Parameter DRC Fehleranzeige zur Auswahl der Bilddarstellungsfarben &fuuml;r DRC Abstands- und Höhenfehler. Bei Einstellung von Fehlerfarbe erfolgt die Anzeige des Fehlerrechteckes in der über Farbpalette eingestellten Fehlerfarbe (standardmässig weiß). Bei Lagenhighlight erfolgt die Darstellung des Rechteckes in der gehighlighteten Lagenfarbe des den Fehler verursachenden Elementes. Dabei werden Fehler auf Lagen, die in der Farbpalette ausgeblendet sind, auch nicht dargestellt.

Das Menü Utilities enthält die Funktion DRC Fehlerliste zur Auflistung der auf dem aktuell geladenen Plan vorhandenen DRC Abstands- und Höhenfehler. Die DRC-Fehlerliste erscheint in einem Popup-Fenster und enthält jeweils Typ, betroffene Lage und Koordinaten des Fehlers. Durch Selektion eines Eintrags der Fehlerliste mit der linken Maustaste kann ein Zoom Fenster auf die Fehlerposition durchgeführt werden. Über die Tasten + und - ist ein Hin- und Herspringen zwischen den einzelnen Fehlerpositionen möglich.

Die in **BAE HighEnd** integrierte Datenstruktur für die Connectivity im **Layouteditor** erlaubt eine *selektive* Kurzschlussanzeige, d.h. bei Kurzschlüssen zwischen zwei Netzen werden nur die tatsächlich den Kurzschluss verursachenden Elemente durch Highlight angezeigt, während in **BAE Professional** der gesamte vom Kurzschluss betroffene Verbindungsbaum gehighlightet wird.

Durch Betätigung der Leertaste gelangen Sie wieder in die Menüoberfläche.

4.6.2 Farbauswahl, Farbtabellen, Vorzugslage

Die Funktion Farbpalette aus dem Menü Ansicht aktiviert ein Popupmenü zur Definition der aktuellen Farbeinstellungen. Durch die Aktivierung dieses Popupmenüs erfolgt gleichzeitig eine Visualisierung der aktuell definierten Farbzuordnungen. Im Popupmenü zur Farbauswahl erfolgt die Zuweisung einer Farbe an einen speziellen Anzeigeelementtyp durch Selektion des Anzeigeelements (bzw. der Lage) über die linke Maustaste sowie die anschließende Selektion der gewünschten Farbe. In den Farbauswahlmenüs des Layoutsystems besteht zusätzlich die Möglichkeit der schnellen Lagen-Ein/Ausblendung mit Erhalt der aktuell eingestellten Farbe. Die Aktivierung bzw. Deaktivierung der Lagenanzeige erfolgt dabei durch Anwahl des Farbbuttons der gewünschten Lage mit der rechten Maustaste. In der Menüanzeige werden die Farbbuttons der aktuell ausgeblendeten Lagen durchgestrichen dargestellt.

Bei Überlappungen verschiedener Elemente werden grundsätzlich die resultierenden Mischfarben dargestellt. Die mit Highlight gewählte Farbe wird ebenfalls mit der Farbe des zu markierenden Elements gemischt und ergibt dann die neue hellere Elementfarbe.

Die aktuell definierte Farbeinstellung kann über das Menü Ansicht mit der Funktion Farben speichern als Farbtabelle unter einem beliebigen Elementnamen (in der Datei **`ged.dat`** im BAE-Programmverzeichnis) abgespeichert werden. Beim Aufruf des Layoutsystems wird automatisch die Farbtabelle mit dem Namen **`standard`** geladen. Jede andere in **`ged.dat`** gespeicherte Farbtabelle lässt sich mit der Funktion Farben laden vom Menü Ansicht je nach Bedarf wieder laden.

Empfehlenswert ist die Definition spezieller Farbtabellen für immer wiederkehrende Arbeitsschritte. So ließe sich z.B. für die Bibliothekserstellung eine Farbtabelle **`stackdef`** (mit Sichtbarkeit der Bohrungen, Bohrplaninformation, usw.) definieren, oder zum schnellen Auffinden noch nicht realisierter Verbindungen eine Farbtabelle **`unroutes`** (in der nur die Airlines sichtbar sind). Sie sollten bei der Definition und jeweiligen Verwendung der Farbtabellen auch darauf achten, dass natürlich der Bildaufbau (insbesondere bei großen Layouts) umso länger dauert, je mehr grafische Objekte durch das System dargestellt werden müssen. Daher sollten Sie die für bestimmte Bearbeitungsphasen zu benutzenden Farbtabellen so definieren, dass möglichst nur die für die Bearbeitung notwendige Information sichtbar ist.

Die bereits wiederholt beschriebene Funktion zur Definition der Vorzugslage im Menü Ansicht bewirkt das automatische Laden einer speziell benannten Farbtabelle. Tabelle 4-3 zeigt die Zuordnung der Farbtabellen-Namen zu den Vorzugslagen sowie die in speziellen Lagenauswahlmenüs wahlweise zulässigen Kurzbezeichnungen zur Spezifikation der Lage über Tastatur (**`<n>`** ist jeweils zu ersetzen durch die Lagennummer).

Tabelle 4-3: Vorzugslagen-Farbtabellen und Lagen-Kurzbezeichnungen

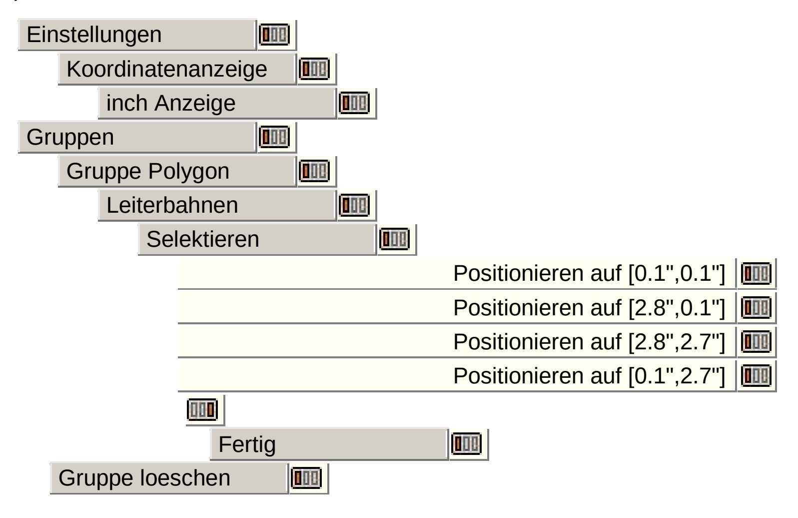
Lage	**Vorzugslagen-Farbtabellenname**	**Lagen-Kurzbezeichnung**
Signallage <n>	layer_<n>	<n>
Signallage Alle Lagen	layer_all	a
Signallage Innenlagen	layer_def	m
Signallage Oberste Lage	layer_def	
Versorgungslage <n>		p<n>
Dokumentarlage <n> Seite 1	layer_d<n>_1	d<n>s1
Dokumentarlage <n> Seite 2	layer_d<n>_2	d<n>s2
Dokumentarlage <n> Beide Seiten	layer_d<n>_a	d<n>sa
Umrandung		b
Unroutes		u

Bei Selektion einer Vorzugslage (z.B. Signallage 2) wird (sofern definiert) die Farbtabelle mit dem zugeordneten Farbtabellen-Namen (z.B. **`layer_s2`**) automatisch geladen. Dies kann z.B. beim interaktiven Routen sehr hilfreich sein.

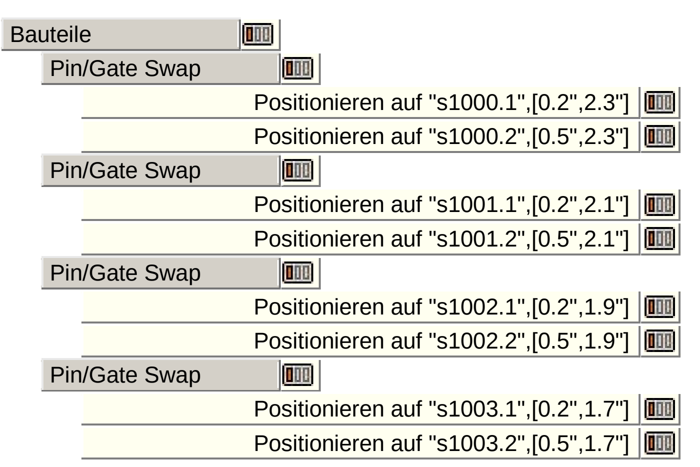
4.6.3 Netzlistenänderungen im Layout

Im **Layouteditor** des **Bartels AutoEngineer** besteht die Möglichkeit, Pin- und Gattertausch durchzuführen, um dadurch eine Vereinfachung des Entflechtungsproblems zu erzielen. Auch können Bauteilnamen geändert werden, um z.B. eine bessere Lesbarkeit des Bestückungsplans zu erreichen. Schließlich besteht im Layout noch die Möglichkeit, alternative Gehäusebauformen zuzuweisen. Alle diese Modifikationen stellen Netzlistenänderungen dar, die selbstverständlich anschließend per Backannotation (siehe auch Kapitel 3.3) in den Stromlauf zurückgemeldet werden müssen.

Im Folgenden sollen einige Pin/Gate Swaps durchgeführt und einige Bauteilnamen geändert werden. Löschen Sie hierzu zunächst (mit Hilfe der Gruppenfunktion) alle Leiterbahnen aus dem Layout (stellen Sie zuvor noch die Koordinatenanzeige auf Inch):



Führen Sie nun jeweils einen Tausch der Pins **1** und **2** für die Schalter-Bauteile **s1000**, **s1001**, **s1002** und **s1003** durch:



Zur Hilfestellung markiert die Funktion Pin/Gate Swap die möglichen bzw. zulässigen Tauschpositionen. Nach Aktivierung der Funktion werden alle Pins, für die in der logischen Bibliothek Swap-Optionen definiert sind, durch eine Kreislinie von 0.6mm Durchmesser in der Arbeitsbereich-Farbe gekennzeichnet. Nach Selektion des ersten Tauschpins wird dieser mit einem gefüllten Quadrat markiert, die für diesen Pin möglichen Tauschpins werden mit einer Kreislinie gekennzeichnet. Der Typ des Tausches wird innerhalb des Kreises durch einen Kennbuchstaben signalisiert. **P** steht für einen Pintausch, **G** für einen Gattertausch und **A** (Array) für einen Gattergruppentausch.

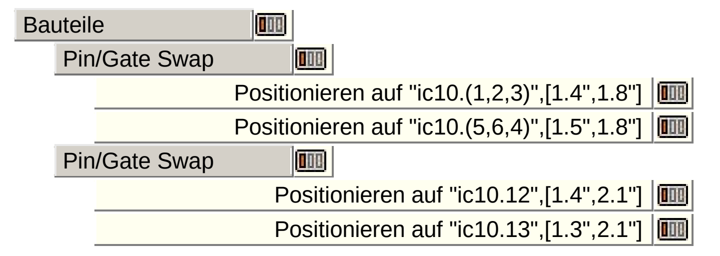
Durch Ausblenden der Arbeitsbereich-Farbe lässt sich die Anzeige der Tauschpositionen unterdrücken.

Jeder Pin- und Gattertausch erfolgt kontrolliert mit Hilfe der in der logischen Bibliothek eingetragenen Vertauschbarkeits-Definitionen (siehe auch Kapitel 7.11, **LOGLIB** und Kapitel 3.2, **Packager**). Ist kein Swap erlaubt, dann gibt die Funktion Pin/Gate Swap die Meldung

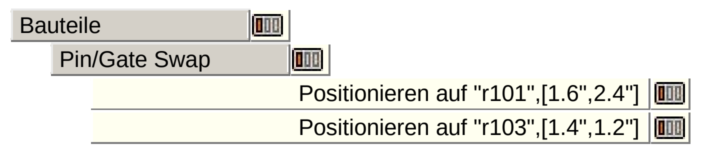
```
Swap zwischen diesen Pins nicht moeglich!
```

aus.

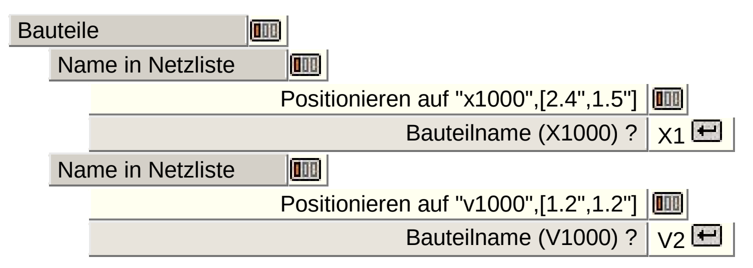
Tauschen Sie nun die Gatter (1,2,3) und (5,6,4) sowie anschließend die Pins `12` und `13` des Bauteils `IC10`:



Führen Sie einen *bauteilübergreifenden* Swap für die Widerstände `R101` und `R103` durch (dies ist nur möglich, da für diese beiden Bauteile dieselben Attributwerte eingetragen sind):



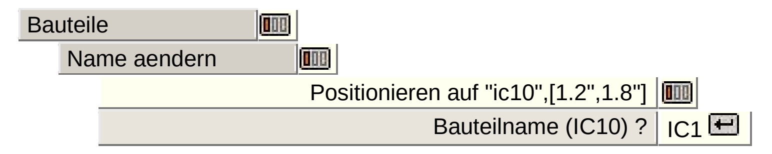
Als nächstes sollen mit Hilfe der Funktion Name in Netzliste einige Bauteilnamen geändert werden. Ändern Sie den Namen des Steckers `X1000` in `X1` sowie den Namen der Diode `V1000` in `V2`:



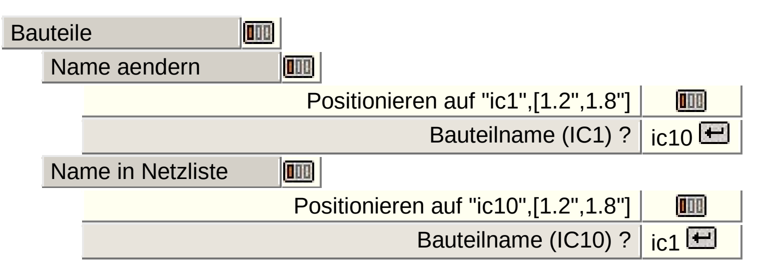
Wenn Sie versuchen, einen Namen zu vergeben, der bereits für ein anderes Bauteil verwendet wurde, dann gibt die Funktion Name in Netzliste die folgende Meldung aus:

```
Der Bauteilname existiert bereits!
```

Ändern Sie nun mit Hilfe der Funktion Name aendern den Namen des Bauteils `IC10` in `IC1`:



Die Airlines zum jetzt mit `IC1` benannten Bauteil sind nun verschwunden. D.h., die Funktion Name aendern führt nicht nur eine Änderung des Bauteilnamens, sondern gleichzeitig auch einen *Bauteiltausch* (bei Beibehaltung der Gehäusebauform) durch. In obigem Beispiel wurde das Bauteil `IC10` gegen das (nicht in der Netzliste befindliche) Bauteil `IC1` ausgetauscht. Machen Sie diesen Bauteiltausch mit der Funktion Name aendern wieder rückgängig, und ändern Sie den Bauteilnamen mit der Funktion Name in Netzliste (anschließend werden die Airlines zum Bauteil `IC1` angezeigt):

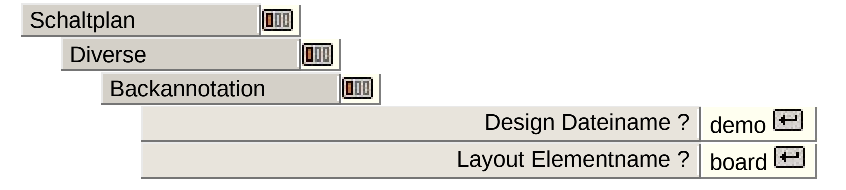


Verwenden Sie die Funktion Name aendern mit aller Vorsicht. Bei fehlerhafter Mehrfachanwendung der Funktion kann es auf bereits gerouteten Layouts ungewollt zu Kurzschlüssen kommen, und die getauschten Bauteile sind nur mehr relativ schwer zu finden.

Die in den vorigen Arbeitschritten durchgeführten Netzlistenänderungen müssen mit Backannotation in den Stromlauf, d.h. in die logische Netzliste zurückgemeldet werden. Speichern Sie hierzu zunächst das Layout, und springen Sie in das Hauptmenü:



Sie befinden sich nun in der BAE-Shell. Wechseln Sie nun mit den folgenden Kommandos in den **Schaltplaneditor**, und starten Sie einen Backannotation-Lauf für das Layout **`board`** aus der Projektdatei **`demo.ddb`**:

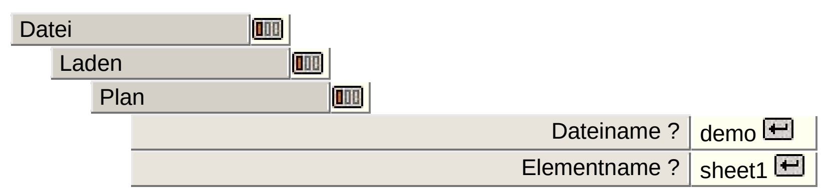


Das System sollte nun folgende Meldungen auf dem Bildschirm ausgeben:

```
==============================
BARTELS BACKANNOTATION UTILITY
==============================

Design Dateiname ........: 'demo'
Layout Elementname ......: 'board'
Es wurden keine Fehler festgestellt.
```

Die Meldung **`Es wurden keine Fehler festgestellt.`** gibt an, dass der Backannotation-Lauf erfolgreich durchgeführt wurde. Sie gelangen nun durch Betätigung einer beliebigen Taste wieder in das Hauptmenü des **Schematic Editors**. Laden Sie nun das Stromlaufblatt **`sheet1`** der Projektdatei **`demo.ddb`**:

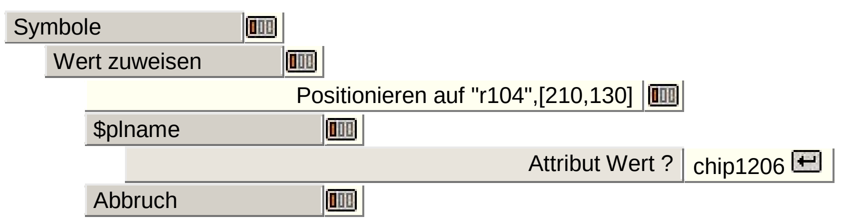


Überprüfen Sie den aktuell geladenen Stromlauf auf die durch Backannotation eingetragenen Änderungen. Achten Sie insbesondere auf die geänderten Bauteilnamen (**`IC1`** anstelle **`IC10`**, **`V2`** anstelle **`V1000`**, ...) und Pinbelegungen (z.B. an den zu **`IC1`** gehörigen Gattern oder an den Schaltern **`S1000`** bis **`S1003`**).

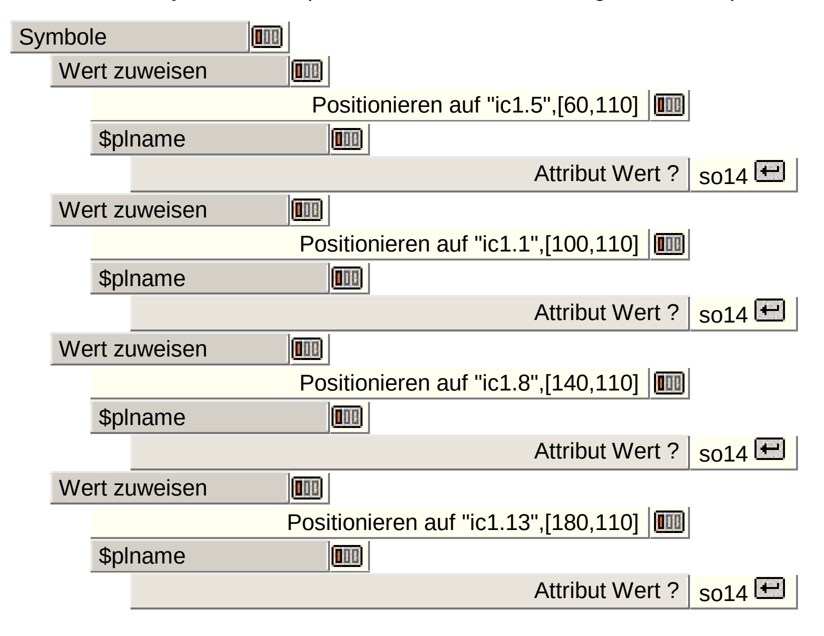
4.6.4 Änderungen im Stromlauf, Redesign

Selbstverständlich besteht die Möglichkeit, ausgehend vom Stromlaufplan die Änderung eines bereits fertiggestellten Designs durchzuführen, ohne ein komplett neues Layout erstellen zu müssen. Hierzu sollen einige Manipulationen in dem im vorigen Arbeitsschritt geladenen Stromlauf vorgenommen werden.

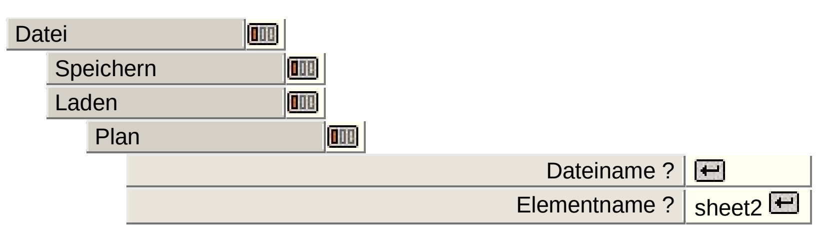
Ändern Sie für den Widerstand **`R104`** den Werteintrag des Attributs **`$plname`** von **`minimelf`** in **`chip1206`**:



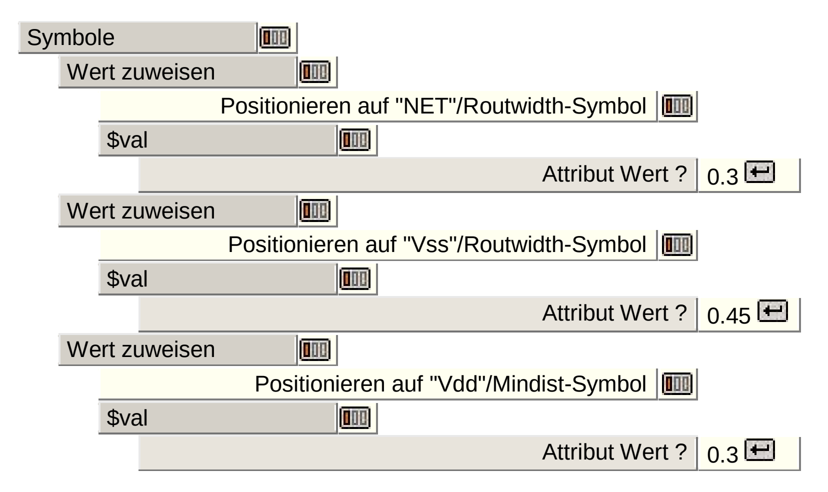
Mit obigem Attributwerteintrag wurde die Gehäusezuweisung für das Bauteil **`R104`** geändert. Nehmen Sie in gleicher Weise eine Änderung der Gehäusezuweisung für *alle* vier Gatter des Bauteils **`IC1`** durch die jeweilige Eintragung des Wertes **`so14`** für das Attribut **`$plname`** vor (Default-Gehäusezuweisung war **`dil14`**):



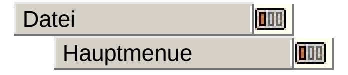
Speichern Sie nun das aktuell geladene Element, und laden Sie **`sheet2`** des Stromlaufs:



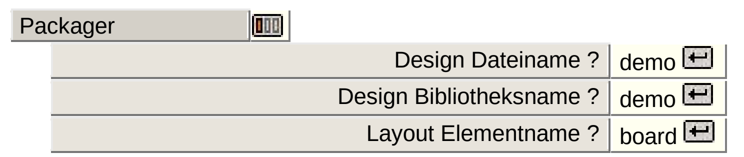
Auf dem aktuell geladenen Schaltplanplan sind einige Router-Steuerparameter (Netzattribute) eingetragen, die modifiziert werden sollen. Ändern Sie die **`ROUTWIDTH`** für das Signal **`NET`** von 0.5mm in 0.3mm, die **`ROUTWIDTH`** für **`Vss`** von 0.6mm in 0.45mm, sowie die **`MINDIST`** für **`Vdd`** von 0.4mm auf 0.3mm:



Springen Sie nun zurück in das Hauptmenü; das aktuell geladene Element wird dabei automatisch gesichert:



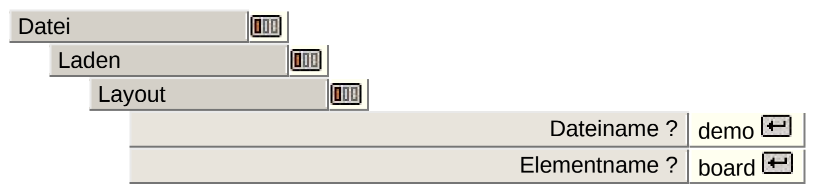
Sie befinden sich nun in der BAE-Shell, von wo aus Sie den **Packager** aufrufen können, um die im Stromlauf der Projektdatei **`demo.ddb`** durchgeführten Änderungen in das Layout (**`board`**) zu übertragen (als Bibliothek kann die Projektdatei selbst benutzt werden, da diese alle für den **Packager**-Lauf notwendigen Einträge enthält):



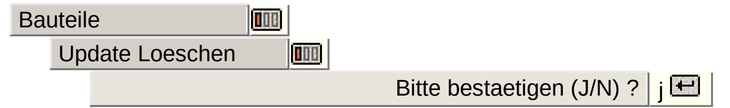
Nach erfolgreichem **Packager**-Lauf (Meldung **`Es wurden keine Fehler festgestellt.`**) gelangen Sie durch Betätigung einer beliebigen Taste wieder in das Hauptmenü, von wo aus Sie den **Layouteditor** aufrufen können:



Laden Sie nun das soeben annotierte Layout:



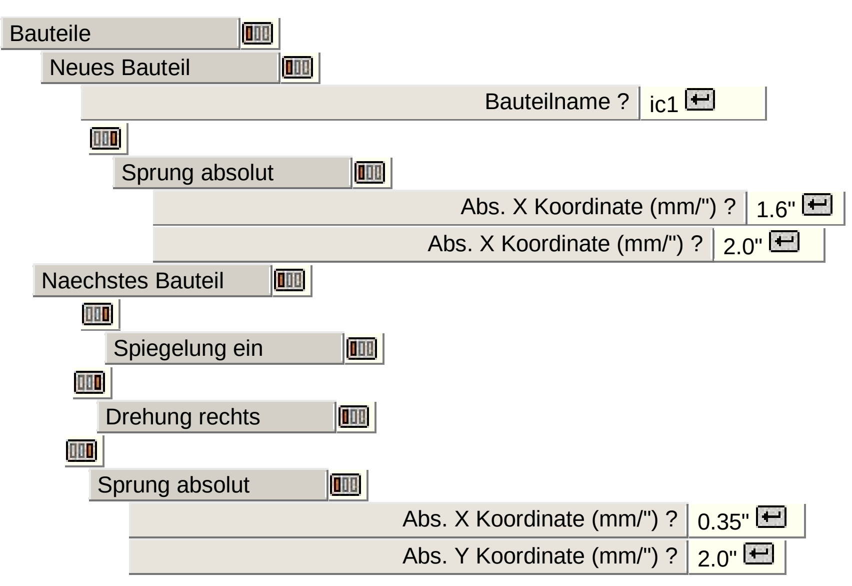
Der **Layouteditor** führt zunächst eine Connectivity Generierung durch. Da die Gehäusebauformen für die beiden Bauteile **`IC1`** und **`R104`** geändert wurden, sind keine Airlines mehr zu diesen beiden Bauteilen eingetragen. Um die seit der letzten Netzlistenmodifikation geänderten Gehäusebauformen aus dem Layout zu löschen, sind folgende Kommandos auszuführen:



Die Funktion Update Loeschen entfernt die beiden Bauteile **R104** und **IC1** aus dem Layout und gibt in der Statuszeile die Meldung

```
Es wurden 2 Bauteile geloescht!
```

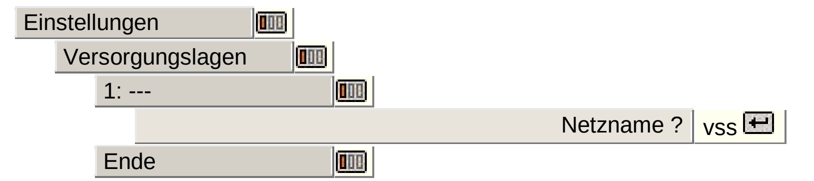
aus. Sie können nun die beiden soeben gelöschten Bauteile mit deren neu zugewiesenen Gehäusebauformen wieder auf dem Layout platzieren:



Nun sind wieder alle Bauteile auf dem Layout platziert. Sie können dies mit der Funktion Naechstes Bauteil überprüfen; das System sollte hierbei die Meldung **Alle Bauteile sind bereits platziert!** ausgeben.

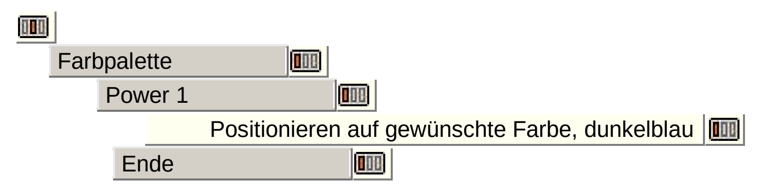
4.6.5 Definieren und Editieren von Versorgungslagen

Das Layoutsystem des **Bartels AutoEngineer** erlaubt die Definition von Versorgungslagen, d.h. sogenannter Power Planes. Definieren Sie mit den folgenden Kommandos eine Versorgungslage für das Signalnetz **Vss**:



Die obige Versorgungslagen-Definition bewirkt, dass Airlines zu den gebohrten Bauteilanschlüssen des Netzes **Vss** entfernt werden. Die Anschlüsse an die Versorgungsinnenlage sind über die gebohrten Pins bereits realisiert. Die Airlines zu den SMD-Anschlüssen des Signalnetzes **Vss** sind hingegen noch vorhanden, d.h. der **Autorouter** wird diese Anschlüsse automatisch (wenn nötig über Vias) an die Versorgungslage anschließen.

Die Versorgungslagen können im Layoutsystem des **Bartels AutoEngineer** visualisiert werden. Überprüfen Sie dies, indem Sie die Farbe für die Versorgungslage 1 auf dunkelblau setzen:



Bei der Darstellung der gebohrten Bauteilanschlüsse erfolgt eine Unterscheidung dahingehend, ob ein Anschluss in die Versorgungslage vorliegt oder nicht. Anschlüsse in die Versorgungslage werden als unausgefüllte Kreise angezeigt, während die zu isolierenden d.h. die nicht zu kontaktierenden Bohrungen als gefüllte Kreise dargestellt werden.

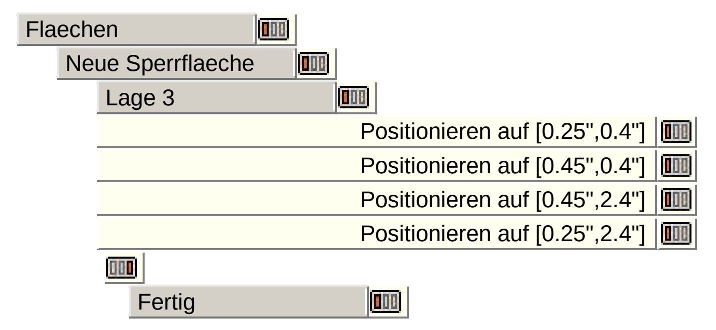
Die Funktion Potentialflaeche aus dem Menü Flaechen ermöglicht über eine entsprechende Lagenauswahl die Definition von Potentialflächen bzw. Isolationsbereichen (Netzname **-**) auf Versorgungslagen. Damit können geteilte Potentiallagen (Split Power Planes) definiert werden. Die Darstellung von auf Versorgungslagen definierten Potentialflächen erfolgt durch die Anzeige ihrer Outline, welche als Isolationslinie zum Rest der Versorgungslage zu betrachten ist und vom **CAM-Prozessor** entsprechend geplottet wird (siehe hierzu auch Kapitel 4.7.6 dieses Handbuchs). Bei der Definition von Potentialflächen auf Versorgungslagen besteht die Einschränkung, dass sich keine zwei Potentialflächen wechselseitig überlagern dürfen, da sonst die Definition der Versorgungslagenbereiche nicht mehr eindeutig ist; in solchen Fällen werden vom Design Rule Check entsprechende Versorgungslagenfehler gemeldet, die auch als eigener Eintrag im Report (Menü Diverse) angezeigt werden. Zulässig ist hingegen die Platzierung von Potentialflächen, die sich komplett innerhalb anderer Potentialflächen befinden; der Anschluss erfolgt in einem solchen Fall jeweils an die "innerste" Potentialfläche.

Die Funktion Neuer Text aus dem Menü Texte, Bohrungen ermöglicht über eine entsprechende Lagenauswahl die Platzierung von Texten auf Versorgungslagen. Damit kann Dokumentation in Form von Textinformation auf Versorgungslagen aufgebracht werden. Bei der Platzierung von Texten auf Versorgungslagen führt der Design Rule Check eine Abstandskontrolle zu den Speziallagen Alle Lagen und Innenlagen durch und erzeugt ggf. entsprechende Abstandsfehlermeldungen. Texte auf Versorgungslagen sind auf der fertig hergestellten Leiterkarte selbstverständlich nur dann sichtbar, wenn die entsprechende Versorgungslage als Außenlage konfiguriert ist. In der Regel jedoch werden Versorgungslagen als Innenlagen definiert, wobei die darauf definierten Texte lediglich als Kontrollinformation (z.B. für den Layoutbearbeiter oder für die Plot- bzw. Filmarchivierung) dienen können.

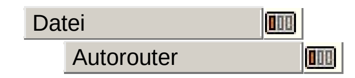
4.6.6 Via-Sperrflächen für den Autorouter

Oft besteht die Notwendigkeit, dem **Autorouter** Flächen vorzugeben, in welchen er keine Durchkontaktierungen setzen, aber dennoch Leiterbahnen führen darf (z.B. unter speziellen Bauteilen). Diese Sperrflächen definiert man am einfachsten auf einer Router-Signallage, die für das eigentliche Layout nicht benötigt wird. Da Vias grundsätzlich über alle Lagen definiert sind (Bohrungen!), erkennt der **Autorouter** den definierten Bereich als Via-Sperrfläche für alle Lagen an. Voraussetzung ist allerdings, dass die Signallage, auf der sich die Sperrfläche befindet, im **Autorouter** als Sperrlage definiert wird, damit der Router die auf dieser Lage befindlichen Elemente bei der Bearbeitung berücksichtigt.

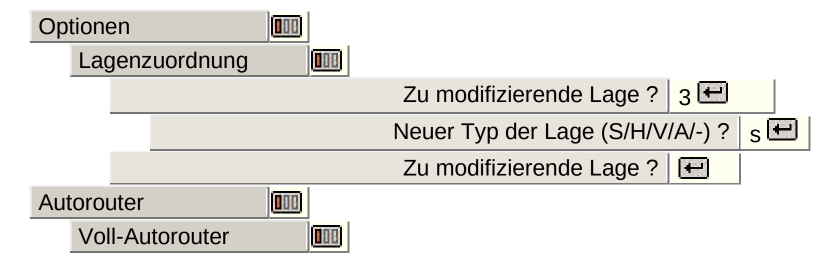
Definieren Sie auf der Signallage 3 eine Sperrfläche unter den Schaltern **`s1000`** bis **`s1009`**:



Rufen Sie nun den **Autorouter** auf:

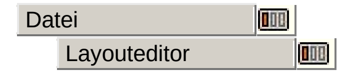


Definieren Sie die Lagenzuordnung so, dass die Signallage 3 vom Router als Sperrlage betrachtet wird, und starten Sie den Vollautorouter (Router-Auflösung, Standard-Leiterbreiten und Standard-Mindestabstände wurden im ersten **Autorouter**-Lauf bereits definiert):



Der Voll-Autorouter sollte das Layout zu 100% entflechten und dabei keine Vias im Bereich der definierten Via-Sperrfläche setzen. Auch die Anschlüsse in die vorher definierte Versorgungslage für das Signalnetz **`Vss`** (siehe oben) werden automatisch erzeugt (für SMD-Anschlüsse ggf. durch ein entsprechendes Leiterbahnstück mit Durchkontaktierung).

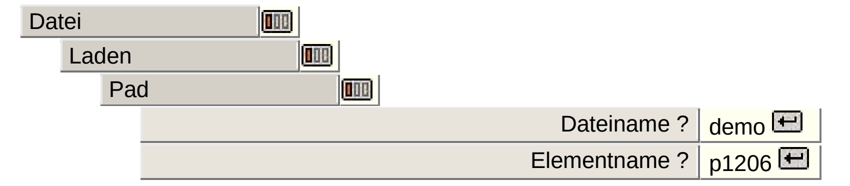
Springen Sie nach erfolgreichem **Autorouter**-Lauf wieder in den **Layouteditor** zurück:



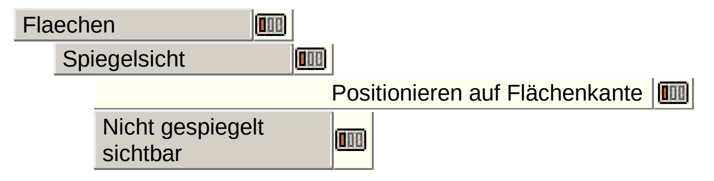
4.6.7 Flächen-Spiegelsicht

Flächen können im **Layouteditor** mit einem Attribut versehen werden, welches angibt, ob die Fläche gespiegelt oder ungespiegelt sichtbar ist. Damit ist es z.B. möglich, an ein und demselben SMD-Pad zwei verschiedene Anschlussflächen zu definieren, von denen die eine gespiegelt (also auf der Lötseite), die andere ungespiegelt (also auf der Bauteilseite) sichtbar ist. Damit lassen sich z.B. komfortabel für Bauteil- und Lötseite unterschiedliche Lötverfahren unterstützen.

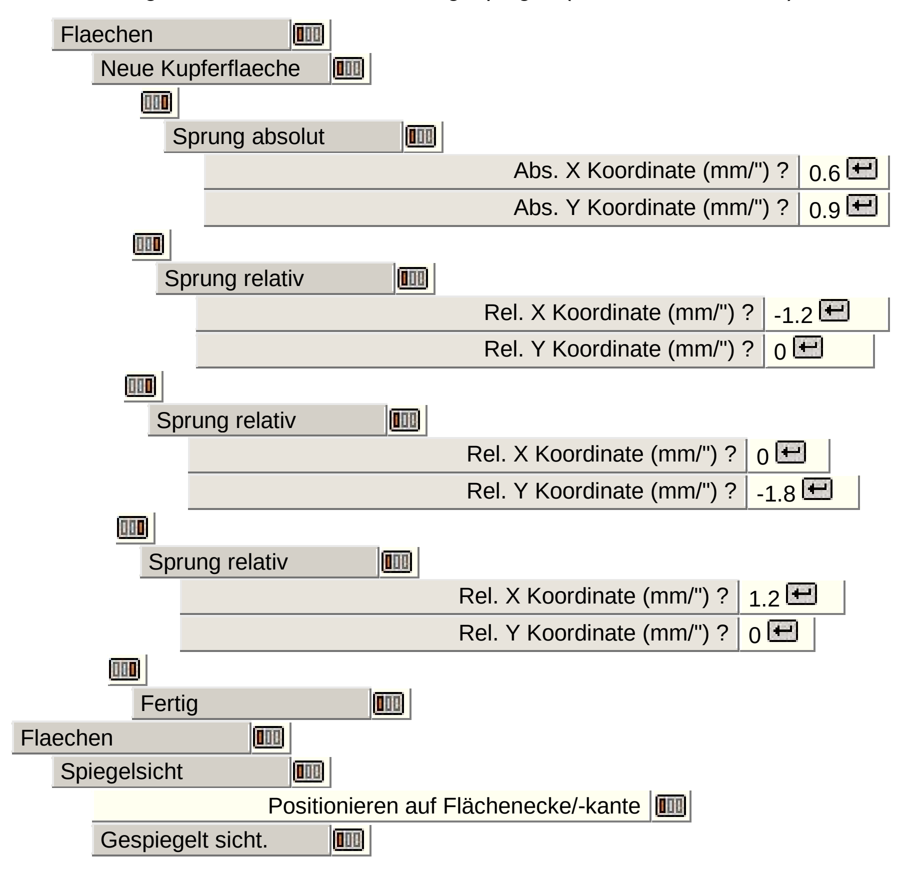
Laden Sie das im Padstack `s1206` für die Gehäusebauform `chip1206` verwendete Padsymbol `p1206` aus der Datei `demo.ddb`:



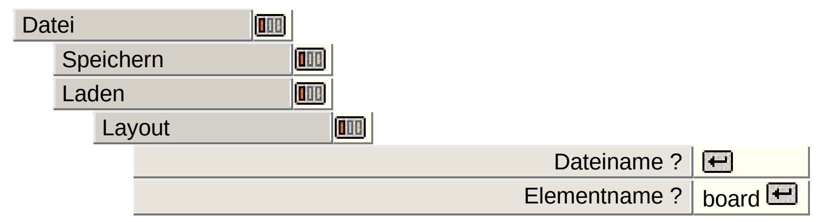
Am geladenen Pad ist eine fingerförmige Anschlussform definiert. Legen Sie mit den folgenden Kommandos fest, dass diese Lötaugenform nur ungespiegelt (also auf der Bauteilseite) sichtbar sein soll:



Die Lötaugenform für die Bauteilseite ist nun definiert. Erzeugen Sie mit den folgenden Kommandos nun eine rechteckige Kupferfläche, und legen Sie fest, dass diese nur gespiegelt (also auf der Lötseite) sichtbar sein soll:



Speichern Sie nun das geänderte Padsymbol ab, und laden Sie wieder das Layout:



Das System führt eine Connectivity Generierung durch, da ein Element aus der Bibliothek der Projektdatei geändert wurde. Vergleichen Sie anschließend die auf Löt- und Bauteilseite unterschiedlichen Lötaugenformen an den Gehäusebauformen `chip1206` (Bauteile `R104` und `C101`, gespiegelt; Bauteil `R105` ungespiegelt).

Die Funktion zur Festlegung einer Spiegelsicht steht grundsätzlich immer bei der Definition von Kupfer-, Sperr-, oder Dokumentarflächen an spiegelbaren Bibliothekselementen zur Verfügung. Durch geeignete Sperrflächendefinitionen auf entsprechenden Dokumentarlagen kann damit z.B. auch die Abstandshaltung von Bauteilen geprüft werden, und zwar auf unterschiedliche Weise je nachdem ob die Bauteile gespiegelt oder ungespiegelt platziert sind.

Unterstützung von Reflow-Reflow-SMD-Lötverfahren

Zur Unterstützung von SMD-Lötverfahren in Reflow-Reflow-Technik steht in der Dialogbox Einstellungen des Menüs Einstellungen der Parameter Flaeche Spiegel-Sicht zur Verfügung. Bei aktivierter Spiegel-Sicht (Voreinstellung) verhält sich das System wie oben beschrieben. Bei Deaktivierung der Spiegel-Sicht, sind unabhängig von der Bauteilspiegelung alle als Nicht gesp. sicht., deklarierten Flächen immer sichtbar und alle als Gespiegelt sicht., deklarierten Flächen nie sichtbar. Mit Hilfe dieser Parametereinstellung kann eine für das konventionelle SMD-Löten erstellte Bibliothek auch für die Reflow-Reflow-Technik werden.

4.6.8 Flächenautomatik

Das BAE-Layoutsystem ist ausgestattet mit Funktionen zum automatischen Füllen von Kupferflächen. Dabei kann mit einstellbarem Isolationsabstand, definierbarer minimaler Strukturgröße, sowie der wahlweisen Elimination isolierter Potentialflächen gearbeitet werden. Wärmefallen werden nach Bedarf automatisch erzeugt, wobei auch die Anschlussbreite einstellbar ist. Elektrisch leitfähige Flächen lassen sich in linien- oder gitterschraffierte Flächen mit definierbarer Schraffurbreite und vorgebbarem Schraffurabstand umwandeln.

Die Funktionen zur Füllflächengenerierung befinden sich im Untermenü Flaechenautomatik des Menüs Flaechen im **Layouteditor**.

Parametereinstellungen für die Flächenautomatik

Abhägig von der aktuellen BAE-Menükonfiguration können die Parameter für die Flächenautomatik entweder über entsprechende Menüfunktionen im Untermenü Flaechenautomatik oder über den Dialog Einstellungen aus selbigem Untermenü spezifiziert werden.

Über den Parameter Isolationsabstand kann der gewünschte Isolationsabstand für die Füflächengenerierung festgelegt werden. Der Defaultwert hierfür beträgt 0.3mm. Abweichend vom eingestellten (Standard-)Isolationsabstand werden mit dem Netzattribut **MINDIST** definierte netzspezifische Mindestabständsvorgaben (siehe hierzu auch Kapitel 7.11 ggf. gesondert berücksichtigt.

Über den Parameter Min. Strukturgroesse kann eine minimale Strukturgröße für die Füllflächengenerierung festgelegt werden. Der Defaultwert hierfür beträgt 0.15mm. Es empfiehlt sich ein Wert in der Größenordnung der kleinsten in der Gerber-Blendentabelle definierte Blende damit später im **CAM-Prozessor** ein Gerber-Plot ohne Überzeichnungsfehler erzeugt werden kann.

Mit Bahnen Aussparung kann spezifiziert werden, ob die abzuisolierenden Leiterbahnen oktagonal (Default-Einstellung) oder rund ausgespart werden sollen.

Über Insel Erkennung kann angegeben werden, ob beim Füllen von Potentialflächen isolierte Teilflächen (sogenannte Inseln) automatisch gelöscht (Inseln loeschen; Standardeinstellung) oder mitgeneriert (Inseln bilden) werden sollen. Die Option Inseln selektieren erzeugt Inselflächen und selektiert die generierten Inselflächen darüberhinaus automatisch zur aktuellen Gruppe.

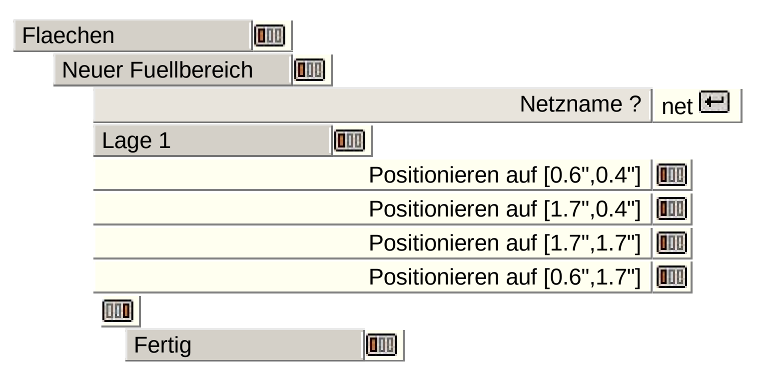
Über den Parameter Waermefallen lässt sich angeben, ob die zur Potentialfläche gehörenden Bauteilanschlüsse und Vias in Form von Wärmefallen oder als Direktanschlüsse ausgeführt werden sollen. Wärmefallen werden mit einer definierbaren Leiterbreite. Die Leiterbreite der Wärmefallenstege (Default 0.3mm) kann frei gewählt werden, und es besteht die Möglichkeit der Spezifikation eines spezifischen Mindestabstands zur Isolation der Wärmefallen (Default 0mm, d.h. Isolation entsprechend der aktuell eingestellten Parameter bzw. Attribute für die Abstandshaltung).

Auch können über entsprechende Optionen wahlweise nur Pins oder nur Vias über Wärmefallen angeschlossen werden. Die entsprechenden Auswahlmöglichkeiten lauten Direktanschluss, Waermefallen Pins & Vias, Waermefallen Pins sowie Waermefallen Vias.

In **BAE HighEnd** ist die Angabe der maximal zulässigen Anzahl von Stegen pro Wärmefalle möglich. Hierbei kann ein Wert von 1 bis 4 angegeben werden. Die Standardsequenz zur Steggenerierung ist links, rechts, unten, oben.

Definition von Flächenfüllbereichen

Die Funktionen zur automatischen Füllflächengenerierung werden auf speziell definierte Arbeitsbereiche für die Flächenautomatik, sogenannte Flächenfüllbereiche angewendet. Definieren Sie mit den folgenden Kommandos auf der Signallage 1 einen rechteckigen Flächenfüllbereich für das Signalnetz `net`:



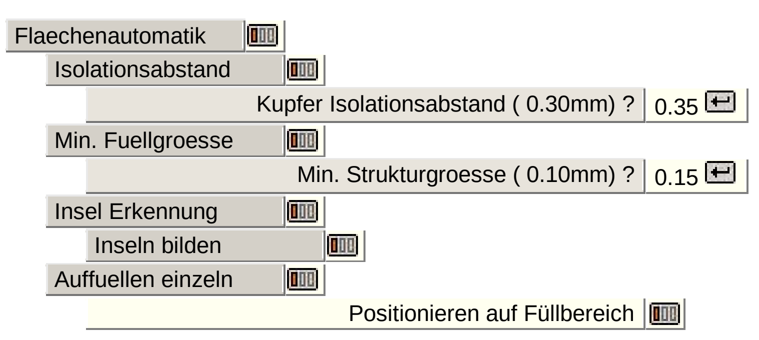
Der Füllbereich wird keinem Potential zugeordnet, wenn auf die Abfrage nach dem Netznamen ein Bindestrich (`-`) eingegeben wird.

Normale Kupferflächen, die an ein Netz angeschlossen sind, werden für das Flächenfüllenwie Potentialflächen dieses Netzes behandelt und werden somit beim Füllen mit dem entsprechendem Netz nicht ausgespart. Dies erleichtert z.B. insbesondere die Handhabung von Teardrops, die als normale Kupferflächen ausgeführt sind.

Füllflächengenerierung

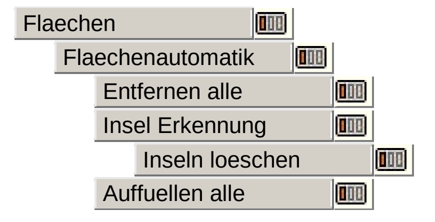
Die Funktionen zur automatischen Füllflächengenerierung stehen im Untermenü Flaechenautomatik des Menüs Flaechen zur Verfügung. Auffuellen alle bewirkt das automatische Füllen aller definierten Füllbereiche. Entfernen alle bewirkt das automatische Löschen der Flächen aus allen definierten Füllbereichen. Bei Auffuellen einzeln bzw. Entfernen einzeln ist der zu bearbeitende Füllbereich entsprechend zu selektieren.

Füllen Sie nun den definierten Füllbereich mit Hilfe der Flächenautomatik auf; stellen Sie den Isolationsabstand dabei auf 0.35mm und die minimale Strukturgröße auf 0.15mm ein, und geben Sie über Insel Erkennung an, dass abisolierte Teilflächen mitgeneriert werden:



Das System erzeugt innerhalb des selektierten Füllbereichs für das Netz `net` Potentialflächen, die von fremdem Potential abgeschirmt werden. Dieser Arbeitsvorgang ist sehr rechenintensiv und kann einige Zeit in Anspruch nehmen. Nach Beendigung des Füllvorgangs empfiehlt es sich, einen Bildneuaufbau zu veranlassen, um das Resultat visuell zu überprüfen. Sind Sie mit dem Ergebnis nicht zufrieden (weil die generierten Strukturen z.B. zu fein oder zu grob sind), dann können sie den Füllvorgang rückgängig machen, neue Flächenfüllparameter einstellen und den Flächenfüll-Algorithmus erneut aufrufen.

Setzen Sie mit den folgenden Kommandos den zuvor durchgeführten Arbeitsschritt zurück, und rufen Sie den Füllalgorithmus anschließend mit automatischer Insel-Elimination auf:



Die minimale Strukturgröße, d.h. die kleinste Flächenausdehnung, mit der gefüllt werden darf, ist von essentieller Bedeutung für die spätere CAM-Ausgabe. Um zu verhindern, dass spitzwinklige Flächenstrukturen entstehen, führt der Füllalgorithmus automatisch eine Abrundung der konvexen Ecken der zu generierenden Flächen entsprechend des gewählten Wertes für die minimale Strukturgröße durch. Für die minimale Strukturgröße sollte demnach z.B. ein Wert etwa im Bereich der Größe der kleinsten in der Gerbertabelle definierten Blende spezifiziert werden. Dadurch ist gewährleistet, dass später im **CAM-Prozessor** ein Gerber-Plot ohne Überzeichnungsfehler erzeugt werden kann.

Komplexitätsbetrachtungen zur Flächenautomatik

Bei der Anwendung der Flächenautomatik kann es je nach Komplexität und Größe der zu füllenden Fläche sowie der abzuisolierenden Strukturen zu längeren Rechenzeiten und größeren Datenmengen kommen. Der Rechenzeitbedarf für die zur Abisolierung notwendige Abstandsberechnung ist bei rechteckigen bzw. orthogonal angeordneten Strukturen sehr viel geringer als z.B. bei kreisbogenförmigen Strukturen, da bei letzteren sehr viel komplexere Geometrieberechnungen anzustellen und damit rechenzeitintensivere Fließpunktoperationen auszuführen sind. Darüber hinaus wirkt sich die Komplexität der abzuisolierenden Strukturen und insbesondere deren Lage zueinander entscheidend auf die durch den Füllalgorithmus erzeugten Datenmengen und die Komplexität der generierten Flächen aus. Abbildung 4-8 verdeutlicht diesen Sachverhalt.

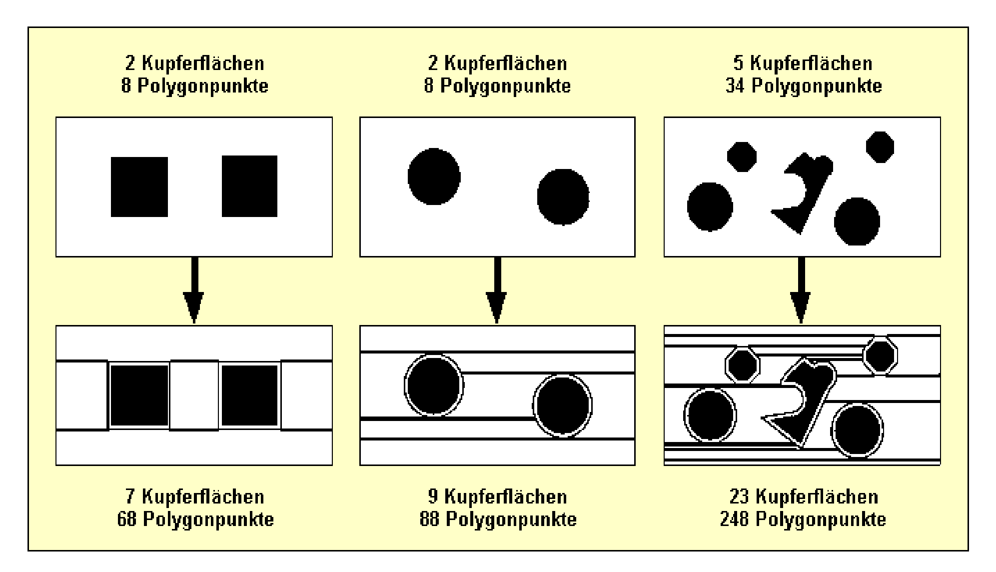


Abbildung 4-8: Flächenautomatik; Komplexitätsbetrachtung

Um die Komplexität und Anzahl der erzeugten Füllflächen zu reduzieren, kann über die Parametereinstellung Bahnenaussparung wahlweise eine der Optionen Oktagon Bahnen, Oktagon Kreise oder Oktagon Bahnen+Kreise anstelle der Defaultoption Runde Ecken aktiviert werden. Damit werden Leiterbahnknicke und/oder Vollkreise für die Aussparung beim Flächenfüllen durch das umschreibende Achteck ersetzt. Durch diese Art der Aussparung lassen sich auch die Gerberdatenmengen reduzieren, wenn für die Gerberausgabe keine Gerber-Kreisbefehle verwendet werden können.

Um zu lange Wartezeiten bei einer versehentlichen Aktivierung der Flächenautomatik zu vermeiden, besteht die Möglichkeit, diese Funktion während der Generierung von Füllflächen vorzeitig per Tastendruck und Bestätigung abzubrechen. Hierbei ist jedoch zu beachten, dass ein Funktionsabbruch in der abschließenden Phase der Connectivity-Generierung nicht mehr möglich ist. Weiterhin ist zu beachten, dass die bis zum Funktionsabbruch generierten Füllflächen ggf. mit der Undo-Funktion explizit zu entfernen sind.

Durch die Erzeugung von Füllflächen kann die Anzahl der Kupferflächen und insbesondere die der Polygonpunkte erheblich zunehmen. Dies hat zur Folge, dass der Rechenzeitbedarf für die Mincon-Funktion zur Berechnung der Unroutes (siehe hierzu auch Kapitel 4.3.2 dieses Handbuchs) enorm zunimmt, wenn die Mincon-Funktion auf eine Ecke-zu-Ecke-Berechnung eingestellt ist. In Extremfällen ist daher die Einstellung der Mincon-Funktion auf eine Pin-zu-Pin-Berechnung empfehlenswert. Der Mincon-Funktionstyp kann über Mincon. Funktion selektiert werden.

Generierung von Schraffurflächen

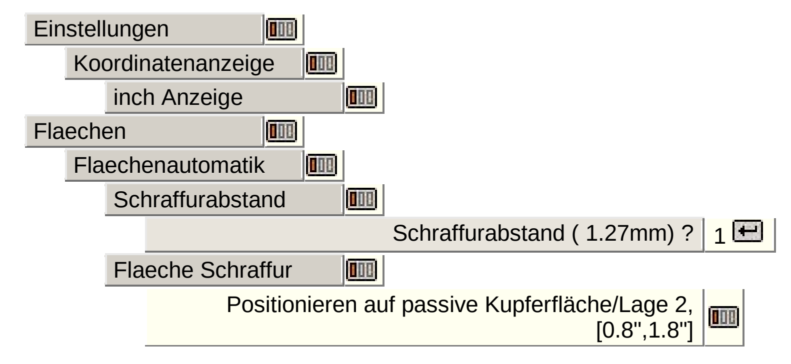
Das Untermenü Flaechenautomatik enthält Funktionen zur automatischen Umwandlung elektrisch leitfähiger Flächen in schraffierte Flächen (Hatching).

Die Schraffur wird durch Leiterbahnzüge generiert. Hierfü können die Leiterbreite und der Abstand zwischen den Leiterbahnen mit Schraffurbreite (Default 0.3mm) und Schraffurabstand (Default 1.27mm) eingestellt werden.

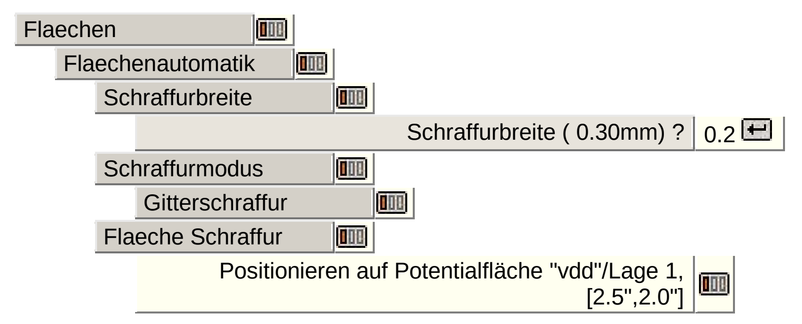
Mit Schraffurmodus kann zwischen unterschiedlichen Schraffurverfahren zur Generierung einer (diagonalen) Linienschraffur oder zur Erzeugung einer Gitterschraffur gewählt werden.

Die Schraffurflächengenerierung wird durch den Aufruf der Funktion Flaeche Schraffur und die Selektion der umzuwandelnden Kupferfläche aktiviert.

Wandeln Sie mit den folgenden Kommandos die auf der Signallage 2 des Layouts platzierte, kreisförmige passive Kupferfläche in eine linienschraffierte Fläche mit einem Schraffurabstand von 1mm um (stellen Sie zuvor die Koordinatenanzeige auf Inch ein):



Setzen Sie nun die Schraffurbreite auf 0.2mm und den Schraffurmodus auf Gitterschraffur, und wandeln Sie die auf der Signallage 1 des Layouts platzierte Potentialfläche für das Signalnetz **vdd** in eine gitterschraffierte Fläche um:



Die Funktion Flaeche Schraffur wandelt das zu bearbeitende Polygon in einen Spezialtyp um. Die Schraffur und die Umrandung der Schraffurfläche werden durch Leiterbahnzüge erzeugt. Diese Leiterbahnen haben die über die Schraffurbreite spezifizierte Leiterbahnbreite und werden der Schraffurfläche fest zugeordnet damit eine Bearbeitung des kompletten Schraffurobjekts mit Standardflächenfunktionen wie Flaeche bewegen oder Flaeche kopieren möglich ist.

Wenn Sie alle Schritte bis hierhin richtig ausgeführt haben, dann sollte das Layout jetzt entsprechend Abbildung 4-9 aussehen.

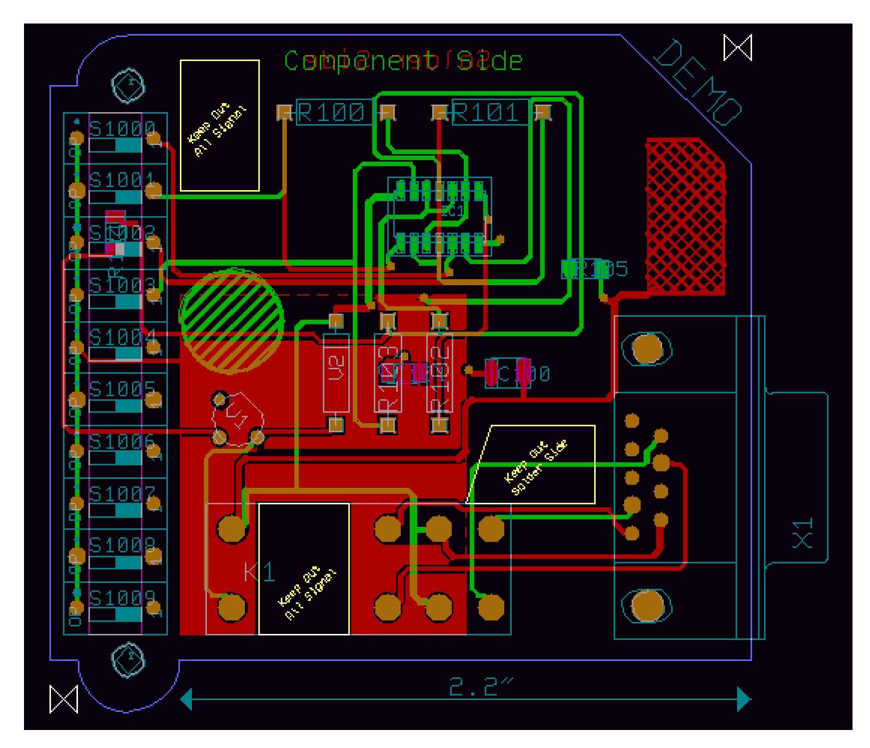
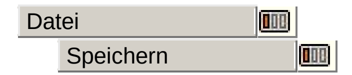


Abbildung 4-9: Layout mit Füllflächen

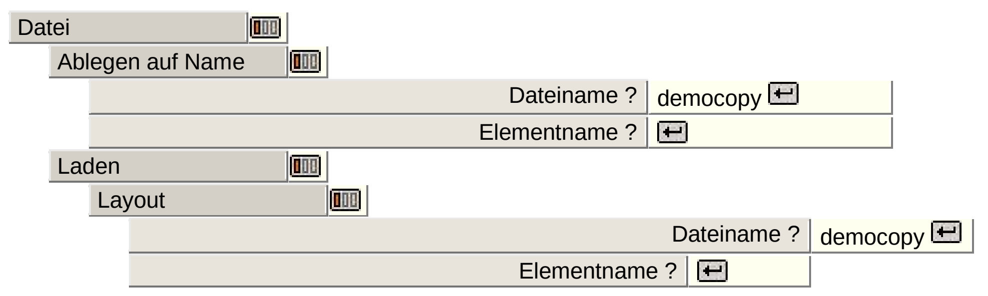
Vergessen Sie nicht, das aktuelle Layout zu sichern:



4.6.9 Bibliotheks-Update

Eines der mächtigsten Werkzeuge des **Bartels AutoEngineer** ist die Funktion Update Bibliothek, mit der in kürzester Zeit eine Aktualisierung der jobspezifischen Bibliothek (in der Regel ist hier eine Angleichung an eine "Master-Bibliothek" verlangt) durchgeführt werden kann.

Kopieren Sie das aktuell geladene Layout in die Datei `democopy.ddb` (mit Default-Elementnamen), und laden Sie das kopierte Layout:



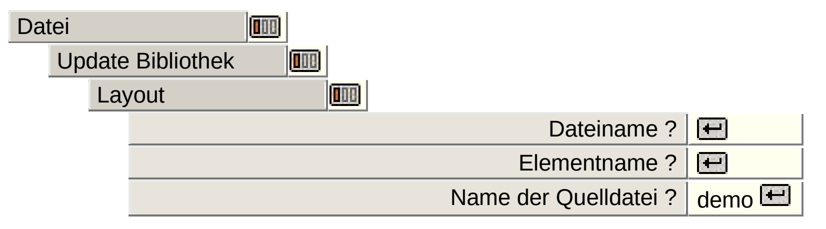
Auf dem Bildschirm sehen Sie nun das kopierte Layout, das allerdings nur die direkt auf dem Layout definierten Objekte enthält. Was fehlt, sind die Bibliothekssymbole (Bauteile, Padstacks, Pads) aus den unteren Hierarchieebenen. Diese wurden durch die Funktion Ablegen auf Name nicht mitkopiert, um zu verhindern, dass evtl. in der Zieldatei `democopy.ddb` bereits vorhandene Elemente überschrieben werden. In der Statuszeile gibt das System die Meldung

```
Einige angeschlossene Pins fehlen!
```

bzw.

```
Einzelne Teile fehlen im Bild (sind nicht ladbar)!
```

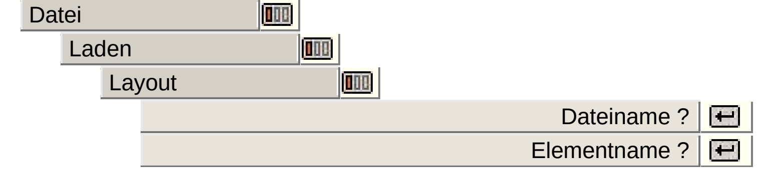
aus. Um nun die jobspezifische Bibliothek der DDB-Datei `democopy` auf die in `demo.ddb` enthaltene Bibliothek anzugleichen, müssen Sie die folgenden Kommandos ausführen:



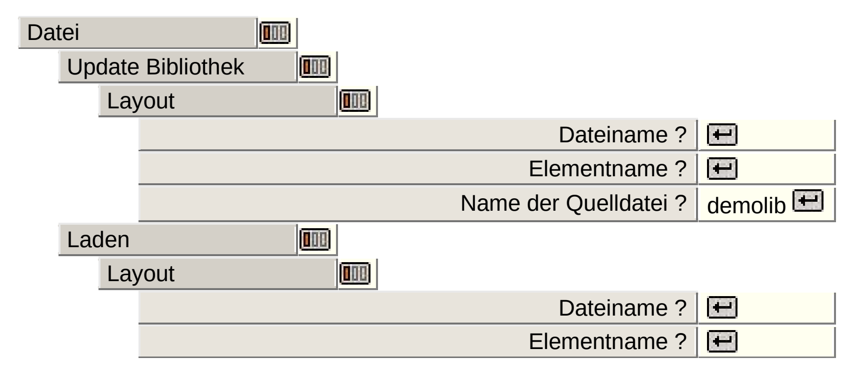
Das System sollte nach wenigen Augenblicken die Meldung

```
Bibliothekselemente ersetzt!
```

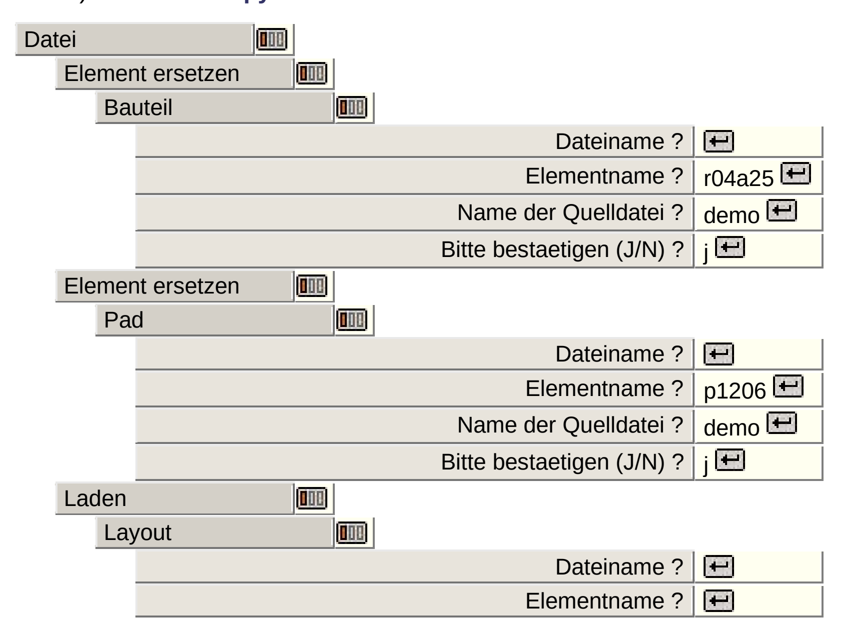
ausgeben. Diese Meldung besagt, dass alle auf dem aktuell geladenen Layout referenzierten Bibliothekselemente aus der Quelldatei kopiert wurden. Um den durchgeführten Bibliotheks-Update zu visualisieren, ist das Layout nun erneut zu laden:



Auf dieselbe Weise kann die jobspezifische Bibliothek des aktuell geladenen Layouts z.B. mit den Einträgen einer anderen Bibliothek korreliert werden. Führen Sie einen Bibliotheks-Update für das im Speicher befindliche Layout mit der Quelldatei **`demolib.ddb`** durch, und achten Sie beim erneuten Laden des Layouts insbesondere auf die bedrahteten Widerstände, deren Gehäusebauform **`r04a25`** in **`demolib.ddb`** anders definiert ist:



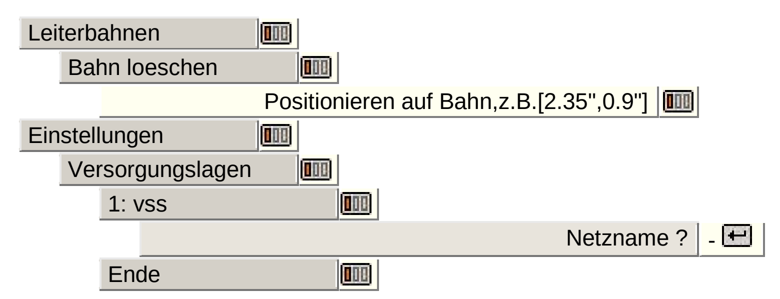
Die Funktion Element ersetzen erlaubt das gezielte Ersetzen spezieller Bibliothekselemente durch entsprechende Elemente aus einer Quelldatei. Mit den folgenden Kommandos können das Bauteilsymbol **`r04a25`** und die Paddefinition **`p1206`** in der (aktuellen) Datei **`democopy.ddb`** wieder ersetzt werden durch die in der Datei **`demo.ddb`** definierten Symbole:



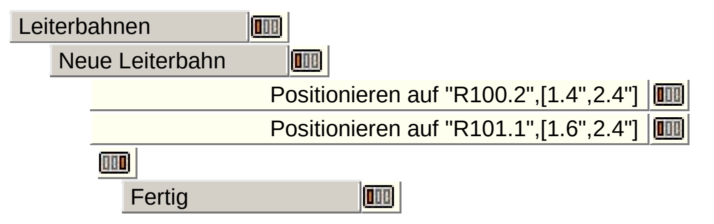
4.6.10 Rücknetzliste

Ausgesprochen nützlich für HF-Anwendungen ist die Funktion Rueck-Netzliste, mit der aus der Information über das auf der Leiterkarte befindliche Kupfer (und *nur* aus diesem) automatisch eine Netzliste erzeugt werden kann.

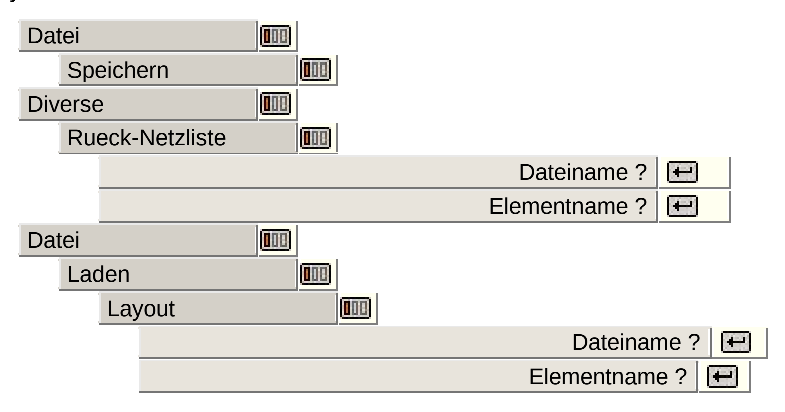
Die Arbeitsweise dieser Funktion soll anhand des aktuell geladenen Layouts (Datei **democopy.ddb**, Default-Elementname) demonstriert werden. Löschen Sie hierzu zunächst die Leiterbahn, die die Pins **NO1** und **4** der Bauteile **K1** und **X1** verbindet sowie die Definition der Versorgungslage **vss**:



Beachten Sie, dass nun Airlines anstelle der gelöschten Leiterbahn und der Versorgungslage angezeigt werden. Verbinden Sie nun mit einer neuen Leiterbahn die beiden Pins **2** und **1** der Bauteile **R100** und **R101** miteinander:



Das System meldet jetzt einen Kurzschluss. Speichern Sie das Layout ab, erzeugen Sie mit Hilfe der Funktion Rueck-Netzliste eine Netzliste, die Sie unter demselben Element- und Dateinamen wie das Layout ablegen, und laden Sie anschließend das Layout wieder:



Das System führt nun eine Connectivity Generierung für das Layout mit der durch Rueck-Netzliste erzeugten Netzliste durch. Es erscheinen anschließend keine Airlines mehr für die zuvor gelöschte Leiterbahn und für das Signalnetz **vss**. Das Layout enthält auch keine offenen Verbindungen oder Kurzschlüsse mehr (überprüfen Sie dies mit Batch DRC und Report im Menü Diverse). Wird die vor dem Aufruf von Rueck-Netzliste neu gelegte Leiterbahn entfernt, dann erscheint an ihrer Stelle eine Airline.

Eine weitere nützliche Anwendung der Funktion Rueck-Netzliste **CAM-View** (siehe hierzu Kapitel 4.8 dieses Handbuchs) auf Signallagen des Layouts eingelesen wurden.

4.6.11 Blind und Buried Vias

Blind und Buried Vias sind Sacklöcher bzw. partielle Durchkontaktierungen, die als Umsteiger beim Routen von Mehrlagen-Layouts verwendet werden können. Durch den Einsatz von Blind und Buried Vias lässt sich unter Umständen der Entflechtungsgrad auf Multilayer-Platinen (mit 4 oder mehr Signallagen) entscheidend erhöhen.

Der **Layouteditor** des **Bartels AutoEngineer** erlaubt die Definition partieller Durchkontaktierungen. Hierzu sind auf Padstackebene die entsprechenden Pads zu laden und auf die gewünschten Lagen zu platzieren (siehe auch Kapitel 4.2.2, Padstackerstellung). Zusätzlich ist eine Bohrung zu definieren, der eine spezifische Bohrungsklasse zugeordnet werden sollte, um bei der CAM-Ausgabe die selektive Ausgabe der an partiellen Durchkontaktierungen definierten Bohrungen zu ermöglichen (siehe auch Kapitel 4.7.13, Ausgabe von Bohrdaten). Die Konventionen für die Spezifikation der Bohrungsklasse sind frei wählbar. Empfehlenswert ist z.B. die Vergabe der Bohrungsklasse **`A`** für Lage 1/2, **`B`** für Lage 2/3, **`C`** für Lage 3/4, usw. Wir bitten zu beachten, dass bei der Erstellung und Verwendung partieller Durchkontaktierungen die layout-spezifische Definition der Obersten Lage (siehe auch Kapitel 4.3.1) von besonderer Bedeutung ist.

Sind z.B. die Padstacks **`via`** (Alle Lagen), **`via_12`** (Umsteiger von Lage 1 nach 2), **`via_23`** (Umsteiger von Lage 2 nach 3) und **`via_34`** (Umsteiger von Lage 3 nach 4) definiert, dann können alle diese Padstacks in die Liste der Durchkontaktierungen selektiert werden und stehen somit für die Entflechtung z.B. eines 4-Lagen-Boards zur Verfügung (siehe hierzu auch Kapitel 4.3.4, Selektion der Durchkontaktierung).

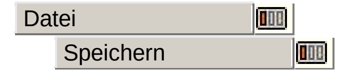
Beim manuellen Routen im **Layouteditor** wird bei jedem Lagenwechsel automatisch das Via mit der geringstmöglichen Lagenbelegung gesetzt. Auch der **Autorouter** arbeitet nach diesem Prinzip. Um allerdings zu verhindern, dass sich während des **Autorouter**-Laufes Mehrdeutigkeiten beim Backtracking ergeben, dürfen - im Gegensatz zum **Layouteditor** - keine zwei Vias in der Via-Liste definiert sein, die sich wechselweise in ihrer Lagenzuordnung in mehr als einer Lage überschneiden. Es ist also z.B. nicht erlaubt, ein Via über die Lagen 1/2/3 und zugleich ein Via über die Lagen 2/3/4 in der Liste der Durchkontaktierungen zu halten; der **Autorouter** bricht in diesem Fall mit der Meldung **`Via Pad-Stack als Via ungeeignet!`** ab. Zulässig hingegen ist z.B. die gleichzeitige Verwendung eines Via über die Lagen 1/2/3, eines Via über die Lagen 1/2 und eines Via über die Lagen 2/3.

Im **CAM-Prozessor** ist bei der Erzeugung der Bohrdaten für die in den partiellen Durchkontaktierungen definierten Bohrungen jeweils die entsprechende Bohrungsklasse zu spezifizieren.

Dringend zu beachten ist, dass bei der Verwendung von Blind und Buried Vias der Aufwand und damit die Kosten für die Fertigung und den Test der Leiterplatte erheblich höher liegen, als bei konventionellen Multilayer-Platinen. Andererseits können die Vorteile spezieller neuartiger Fertigungstechnologien wie etwa des Verfahrens zur Herstellung plasmageätzter Vias nur durch die Verwendung partieller Durchkontaktierungen genutzt werden.

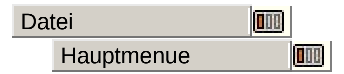
4.6.12 Verlassen des Layoutsystems

Bevor Sie den **Layouteditor** verlassen, sollten Sie nicht vergessen, das gerade bearbeitete *Dateielement* zu *sichern*:



Rücksprung ins Hauptmenü

Von jedem Modul des Layoutsystems (außer **Autorouter**) ist mit den folgenden Kommandos der Rücksprung in das BAE-Hauptmenü möglich:

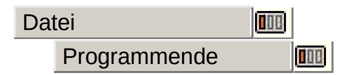


Sie befinden sich anschließend in der Shell des BAE. Diese lässt sich wie folgt beenden:

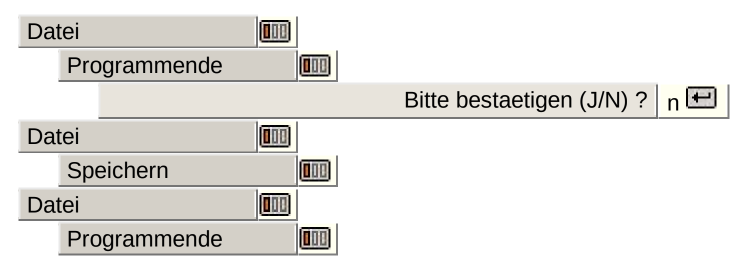


Programmende

Der **Bartels AutoEngineer** kann von jedem Modul des Layoutsystems mit



direkt beendet werden. Sollte der **Layouteditor** hierbei mit einer Bitte um Bestätigung reagieren, dann wurde das aktuell geladene Element noch nicht gesichert. In diesem Fall sollten Sie die Funktion Programmende abbrechen, das aktuell im Speicher befindliche Element sichern und anschließend erst das Programmende herbeiführen:

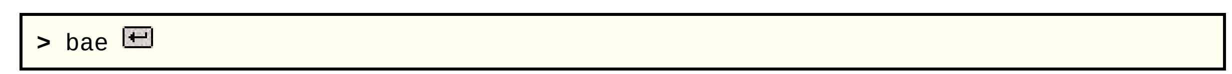


4.7 CAM-Prozessor

4.7.1 Programmaufruf

Bei realen Projekten sollten Sie vor dem Aufruf des **CAM Processors** zur Erzeugung von Fertigungsdaten zuerst immer mit der Funktion Batch DRC aus dem Menü Utilities des **Layouteditors** eine komplette Entwurfsregelprüfung durchführen und das Ergebnis dieses Design Rule Checks mit Report auf etwaige Probleme hin überprüfen. Keinesfalls sollten Sie Fertigungsdaten aus Designs mit Abstandsverletzungen, Kurzschlüssen oder offenen Verbindungen für die Fertigung freigeben, da damit (fast ausnahmlos) unbrauchbare Leiterkarten produzieret werden würden.

Der Aufruf des **Bartels AutoEngineer** sollte grundsätzlich aus dem Verzeichnis erfolgen, in welchem die zu bearbeitenden Projektdateien abgelegt bzw. abzulegen sind. Wechseln Sie also zunächst in Ihr Projektverzeichnis. Zur Abarbeitung der in diesem Handbuch aufgeführten Beispiele ist es zweckmäßig, in das bei der Installation des **Bartels AutoEngineer** angelegte BAE-Jobs-Directory (d.h. in das Verzeichnis, in dem die Datei **`demo.ddb`** abgelegt ist) zu wechseln. Der Aufruf des Layoutmoduls erfolgt aus der Shell des **Bartels AutoEngineer**. Starten Sie diese von Betriebssystemebene aus mit folgendem Befehl:



Wählen Sie den Menüpunkt Layout mit der Maus an, und bestätigen Sie Ihre Wahl durch Drücken der linken Maustaste:

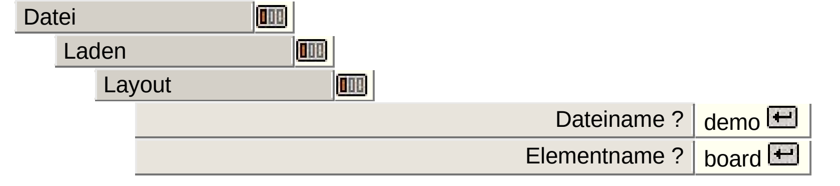


Nun wird der **Layouteditor** des **AutoEngineer** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

Der **CAM-Prozessor** wird mit der Funktion CAM-Prozessor im Menü Datei des **Layouteditors** aktiviert:



War im **Layouteditor** ein Layout geladen, so wird dieses auch automatisch vom **CAM-Prozessor** geladen. Im **CAM-Prozessor** selbst kann man mit der Funktion Laden im Menü Datei ein Layout, ein Bauteil, einen Padstack oder ein Pad zur CAM-Ausgabeverarbeitung laden. Laden Sie mit den folgenden Kommandos das Layout **`board`** aus der Projektdatei **`demo.ddb`** in den Arbeitsspeicher:



Tritt beim Laden eines Elements die Meldung

```
Zeichen Font kann nicht geladen werden!
```

auf, so bedeutet dies, dass der dem Layoutelement zugeordnete Zeichensatz nicht aus der Fontdatendatei **`ged.fnt`** im Programmverzeichnis geladen werden konnte.

4.7.2 Hauptmenü

In der Benutzeroberfläche des **CAM-Prozessors** werden neben bereits aus dem **Layouteditor** bekannten Menüs (wie Ansicht bzw. Bilddarstellung, Diverse) Funktionen zur Generierung von Bestückdaten, Bohrdaten, Gerber-Fotoplots und Kontrollplots (HP-GL, PCL, Postscript) angeboten. Nach dem Aufruf des **CAM-Prozessors** befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden des **CAM-Prozessor** ist das Menü Dateiverwaltung aktiviert, und der grüne Menübalken steht auf Laden.

Unter Windows und Motif kann anstelle der Standard- bzw. Seitenmenükonfiguration wahlweise auch ein Benutzerinterface mit Pulldownmenüs aktiviert werden. Hierzu ist mit Hilfe des Utilityprogramms **BSETUP** das Kommando **`WINMENUMODE`** mit der Option **`PULLDOWN`** in das Setup der BAE-Software einzuspielen (siehe hierzu auch Kapitel 7.2). Bei der Verwendung von Pulldownmenüs ist das Hauptmenü als horizontal ausgerichtete Menüleiste am oberen Ende der Benutzerschnittstelle angeordnet.

Das Hauptmenü ist während der Dauer der Layoutbearbeitung mit dem **CAM-Prozessor** ständig verfügbar und ermöglicht die Aktivierung der folgenden Menüs:



Ansicht, Bilddarstellung

Im Menü Ansicht bzw. Bilddarstellung, das Sie außer durch Selektion im Hauptmenü auch immer über die mittlere Maustaste erreichen können, können Sie Zoomfunktionen aktivieren, das Eingaberaster definieren oder die Farbtabelle einstellen.

Kontrollplot

Das Menü Kontrollplot enthält die Funktionen zur Generierung von HP-GL-Penplots sowie zur Erzeugung von Postscript- und HP-Laser-Ausgaben.

Gerber Fotoplot

Das Menü Gerber Fotoplot enthält die Funktionen zur Generierung von Gerber-Photoplot-Ausgaben. Auch können von hier aus die Gerber-Blendentabellen definiert bzw. verwaltet werden.

Bohr+Bestueckdaten

Das Menü Bohr+Bestueckdaten enthält die Funktionen zur Ausgabe von Bohrdaten im Format Sieb&Meier sowie zu Ausgabe von Bestückdaten in generischem Format.

Plotparameter

Das Menü Plotparameter enthält Funktionen zum Setzen allgemeiner Plotparameter (Symboltoleranz, Nullpunkt, Drehung, Spiegelung) und zur Definition der Parameter für die Ausgabe von Versorgungslagen. Darüber hinaus kann über dieses Menü auch die Ausgabe spezieller Lagen (Alle Lagen, Umrandung, Passermarken) gesteuert werden.

Diverse

Im Menü Diverse kann der Programmabbruch, der Rücksprung in die Shell des **Bartels AutoEngineer** oder der Rücksprung in den **Layouteditor** veranlasst werden. Dieses Menü enthält weiterhin wichtige Dateiverwaltungsfunktionen zum Laden von Elementen und zum Auflisten von Dateiinhalten. Darüber hinaus werden über das Menü Diverse Funktionen zum Definieren des Hintergrundrasters, zum Setzen des Koordinatenanzeigemodus sowie zum Rücksetzen der Fehleranzeige angeboten. Auch der explizite Aufruf von **User Language**-Programmen ist von diesem Menü aus möglich.

4.7.3 Modifizierte Benutzeroberfläche

Menübelegung und Tastaturprogrammierung

Einige der mit der BAE-Software installierten **User Language**-Programme definieren implizite **User Language**-Programmaufrufe über die eine weit reichend modifizierte Benutzeroberfläche mit einer Vielzahl von Zusatzfunktionen (Startups, Toolbars, Menübelegung, Tastaturprogrammierung) aktiviert wird. Das **User Language**-Startupprogramm **BAE_ST** wird automatisch beim Aufruf des **CAM-Prozessors** gestartet. **BAE_ST** ruft seinerseits das **User Language**-Programm **UIFSETUP** auf, welches eine vordefinierte Menü- und Tastaturbelegung im **CAM-Prozessor** aktiviert. Änderungen bzw. Anpassungen der Menü- und Tastaturbelegung können *zentral* in der Quellcodedatei von **UIFSETUP** vorgenommen werden. Die aktuelle Tastaturbelegung kann mit dem **User Language**-Programm **HLPKEYS** angezeigt werden. Der Aufruf von **HLPKEYS** ist über die Funktion Tastaturbelegung aus dem Menü Hilfe möglich, sofern die vordefinierte Menübelegung aus **UIFSETUP** aktiviert ist. Mit dem **User Language**-Programm **UIFDUMP** kann die in der aktuellen Interpreterumgebung definierte Menü- und Tastaturbelegung in Form eines Reports angezeigt bzw. auf eine Datei ausgegeben werden. Mit dem **User Language**-Programm **UIFRESET** lässt sich die komplette Menü- und Tastaturbelegung zurücksetzen. **UIFSETUP**, **UIFDUMP** und **UIFRESET** sind auch über das Menü des **User Language**-Programms **KEYPROG** aufrufbar, welches zudem komfortable Funktionen zur Online-Tastaturprogrammierung sowie zur Verwaltung von Hilfstexten für **User Language**-Programme zur Verfügung stellt.

Kaskadierende Pulldownmenüs unter Windows/Motif

Die Windows- und Motifversionen des **CAM-Prozessors** ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Pulldownmenüs der Windows- und Motifversionen des **CAM-Prozessors** werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs ausgestattet. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

Dialoge für Parametereinstellungen unter Windows/Motif

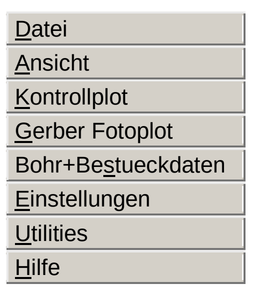
In den Windows- und Motifversionen des **CAM-Prozessors** sind die folgenden Dialoge für Parametereinstellungen implementiert:

- Einstellungen - Einstellungen: Allgemeine CAM-/Plotparameter
- Ansicht - Einstellungen: Bilddarstellungsparameter
- Kontrollplot - Einstellungen: Kontrollplotparameter
- Gerber Fotoplot - Einstellungen: Gerberplotparameter
- Bohr+Bestückdaten - Einstellungen: Bohrdatenausgabeparameter

In den Pulldownmenükonfigurationen werden die Standardfunktionen für Parametereinstellungen über das **User Language**-Programm **UIFSETUP** durch die obigen Menüfunktionen zum Aufruf der entsprechenden Dialoge ersetzt.

Pulldownmenükonfiguration unter Windows/Motif

Bei der Verwendung von Pulldownmenüs unter Windows und Motif wird über das **User Language**-Programm **UIFSETUP** eine an Windows angepasste Menüanordnung mit zum Teil geänderten Funktionsbezeichnungen und einer Vielzahl von Zusatzfunktionen konfiguriert. Das Hauptmenü des **CAM-Prozessors** wird dabei wie folgt aufgebaut:



4.7.4 Grundsätzliches zur Bedienung

Automatische Parametersicherung

Im **CAM-Prozessor** ist eine Funktion zur automatischen Sicherung wichtiger Bearbeitungs- und Plotparameter implementiert. Beim Wechsel in ein anderes BAE-Modul (Hauptmenü oder **Layouteditor**) werden automatisch die folgenden Parameter in der aktuell bearbeiteten Designdatei gespeichert (d.h. die Parametersicherung kann durch Beenden des **AutoEngineer** mit der Funktion Programmende wahlweise auch unterdrückt werden):

- Name der aktuell geladenen Layoutfarbtabelle
- Eingaberaster
- Hintergrundraster
- Koordinatenanzeigemodus
- Breitendarstellungswert
- Ausgabemodus Platinenumrandung
- Ausgabemodus Alle Signallagen
- Ausgabemodus Passermarken-Lage
- Plot/CAM Genauigkeit/Symboltoleranz
- Plot/CAM Nullpunktkoordinaten
- Plot/CAM Rotationsmodus
- Plot/CAM Spiegelungsmodus
- Versorgungslagen Wärmefallenmindestabstand
- Versorgungslagen Isolationsmindestabstand
- Versorgungslagen Wärmefallentoleranz
- Versorgungslagen Isolationstoleranz
- Versorgungslagen Umrandungsbreite
- Versorgungslagen Isolationsbreite
- Kontrollplot Ausgabedateiname/Ausgabeeinheit
- Kontrollplot Massstab/Skalierungsfaktor
- HP-GL-Plot Stiftbreite/Standardlinienbreite
- HP-GL-Plot Geschwindigkeit
- HP-GL-Plot Füllmodus
- Gerber-Fotoplot Ausgabedateiname/Ausgabeeinheit
- Gerber-Fotoplot Blendentabellenname
- Gerber-Fotoplot Format/Plottereinheiten
- Gerber-Fotoplot Standardlinienbreite
- Gerber-Fotoplot Füllverfahren
- Gerber-Fotoplot Kreisbogenausgabemodus
- Bohrdaten Ausgabedateiname/Ausgabeeinheit
- Bohrwerkzeugtabelle Ausgabedateiname/Ausgabeeinheit
- Bohrdaten Werkzeugtoleranz
- Bestückdaten Ausgabedateiname/Ausgabeeinheit

Die Elementnamen der zu sichernden Parametersätze werden vom aktuell bearbeiteten Layoutelement abgeleitet. Layoutspezifische Parametersätze erhalten den Elementnamen des aktuell bearbeiteten Layouts, bauteilspezifische Parametersätze den Namen **`[part]`**, padstackspezifische Parametersätze den Namen **`[padstack]`**, padspezifische Parametersätze den Namen **`[pad]`**. Beim Laden eines Elements wird automatisch der entsprechende Parametersatz mitgeladen. Dadurch wird in komfortabler Weise eine spezifische Arbeitsumgebung zur Bearbeitung der selektierten Bibliothekshierarchie bzw. des selektierten Designobjekts aktiviert.

Ausgabekanal

Bei allen CAM-Ausgabearten kann ein Ausgabekanal (d.h. ein Ausgabegerät bzw. eine Ausgabedatei) angegeben werden, in den die erzeugten CAM-Daten geleitet werden. Dieser Kanal kann entweder direkt ein Ausgabegerät über einen Schnittstellennamen (z.B. COM2 in MS-DOS) oder den Namen einer Datei, in die die Ausgaben gelenkt werden, angeben. Dabei ist zu beachten, dass die Schnittstelle initialisiert ist bzw. Schreibberechtigung und genügend Platten-/Diskettenkapazität zur Aufnahme der Ausgabedaten bereitsteht. Sind diese Grundvoraussetzungen nicht erfüllt, so bricht die Ausgabe mit der Meldung `Schreiben ASCII-Datei fehlgeschlagen!` ab.

Wird kein Ausgabekanal explizit vorgegeben, dann bittet das Programm nach Aktivierung des entsprechenden Ausgabemenüpunktes um die Angabe eines Namens für die Ausgabedatei bzw. den Ausgabekanal. In den Funktionen Plotausgabe Kanal, Plot HP-GL Ausgabe, Postscript Ausgabe und HP Laser Ausgabe im Menü Kontrollplot, in den Funktionen Fotoplotter Kanal, Blenden ausgeben und Plot Gerber im Menü Gerber Fotoplot, sowie in den Funktionen Bandgeraet Kanal, Tabellen-Drucker Kanal, Bohrband erstellen, Werkzeugtabelle erstellen und Bestueckdaten Ausgabe im Menü Bohr+Bestueckdaten sind Popupmenüs zur schnellen Selektion des Ausgabekanals bzw. der Ausgabedatei integriert. Aus Gründen der Datensicherheit werden hierbei Dateien mit den Endungen `.ass`, `.con`, `.ddb`, `.def`, `.exe`, `.fre`, `.ulc` und `.usf` ausgeblendet. Wahlweise besteht selbstverständlich auch die Möglichkeit, die Namen neu zu erstellender CAM-Dateien direkt über Tastatur einzugeben.

User Language

Im **CAM-Prozessor** ist der **Bartels User Language Interpreter** integriert, d.h. vom **CAM-Prozessor** aus können **User Language**-Programme gestartet werden. Der Anwender hat damit die Möglichkeit, eigene Zusatzfunktionen nach anwender- bzw. firmenspezifischen Bedürfnissen zu implementieren und in den **CAM-Prozessor** einzubinden. Hierzu zählen zum Beispiel Statusanzeigen und Parametereinstellungen, Report- und Testfunktionen, spezielle Postprozessoren und Plotfunktionen, firmenspezifische Batch-Prozeduren, usw. usf.

Im **CAM-Prozessor** können **User Language**-Programme explizit oder implizit aufgerufen werden. Der explizite Programmaufruf erfolgt über den Menüpunkt Anwenderfunktion im Menü Datei. Nach der Aktivierung dieses Menüpunktes ist auf die Abfrage nach dem Programmnamen der Name des aufzurufenden **User Language**-Programms (z.B. `ulprog`) explizit einzugeben. Die Betätigung einer beliebigen Maustaste oder die Eingabe eines Fragezeichens **?** auf die Abfrage nach dem Programmnamen bewirkt hierbei die Aktivierung eines Popupmenüs mit allen aktuell verfügbaren **User Language**-Programmen.

User Language-Programme können auch implizit über die Tastatur aktiviert werden. Diese Art des Programmaufrufs ist immer dann möglich, wenn nicht gerade eine andere interaktive Eingabe über Tastatur erwartet wird. Die Spezifikation des Programmnamens erfolgt dabei implizit durch Drücken einer Taste. Zulässige Tasten sind dabei die Standardtasten (1, 2, ..., 0, a, b, c, ...; entsprechende Programmnamen sind **cam_1**, **cam_2**, ..., **cam_0**, **cam_a**, **cam_b**, **cam_c**, ...) bzw. die Funktionstasten (F1, F2, ...; entsprechende Programmnamen sind dabei **cam_f1**, **cam_f2**, ...).

Der **CAM-Prozessor** ermöglicht den ereignisgesteuerten Aufruf von **User Language**-Programmen. Dabei lösen spezielle Ereignisse bzw. Operationen implizit, d.h. automatisch den Aufruf von **User Language**-Programmen mit definierten Namen aus, sofern diese verfügbar sind. Im Einzelnen sind dies die **User Language**-Programme **CAM_ST** beim Starten des **CAM-Prozessors**, **CAM_LOAD** nach dem Laden eines Elements, **CAM_SAVE** vor dem Speichern eines Elements, **CAM_TOOL** bei Selektion eines Toolbarelements sowie **CAM_ZOOM** bei Änderung des Zoomfaktors. Der Aufruf über die Startupsequenz der Interpreterumgebung eignet sich besonders zur automatischen Voreinstellung von modulspezifischen Parametern sowie zur Tastaturprogrammierung und Menübelegung. Der implizite Aufruf von **User Language**-Programmen nach dem Laden bzw. vor dem Speichern von Elementen ermöglicht die automatische Aktivierung elementspezifischer Bearbeitungsparameter wie z.B. des zuletzt selektierten Zoombereichs oder spezieller Farbeinstellungen. Bei Interaktionen in der Werkzeugliste werden die den selektierten Toolbarelementen zugewiesenen Funktionen ausgelöst. Die Änderung des Zoomfaktors kann dazu benutzt werden, Aktualisierungen in Funktionen zur Verwaltung von Entwurfsansichten auszulösen.

Mit der **Bartels User Language** werden darüber hinaus mächtige Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Beachten Sie bitte, dass über die mit der BAE-Software ausgelieferten **User Language**-Programme eine Vielzahl von Zusatzfunktionen implementiert und transparent in die Benutzeroberfläche des **CAM-Prozessors** eingebunden sind.

Eine ausführliche Beschreibung der **Bartels User Language** finden Sie im Bartels User Language Programmierhandbuch (Kapitel 4.2 enthält eine Auflistung aller mit der BAE-Software ausgelieferten **User Language**-Programme).

Neuronales Regelsystem

Im **Bartels AutoEngineer** sind eine Vielzahl mächtiger Zusatzfunktionen über das integrierte **Neuronale Regelsystem** implementiert. Kapitel 6.3.2 enthält eine Übersicht über die im Layoutsystem bereitgestellten Regelsystemanwendungen. Eine Reihe dieser Applikationen ermöglichen spezielle Fertigungsdatenausgaben bzw. eine erweiterte Kontrolle über die Fertigungsdatengenerierung.

4.7.5 Plotparameter

Vor dem Plotten sollte man sich über die Art und gewünschte Genauigkeit der Ausgabe im Klaren sein. Der **CAM-Prozessor** bietet eine Vielzahl von Parametern zur Steuerung der Plotausgabe.

Bei der Angabe von Parametern wird hinter dem Eingabeprompt jeweils in Klammern der aktuelle Wert angezeigt. Fehleingaben werden mit der Meldung **`Ungueltiger numerischer Wert!`** zurückgewiesen, und es wird der angezeigte vorherige Parameterwert beibehalten.

Im Menü Plotparameter des **CAM-Prozessors** sind die für alle Ausgabearten gemeinsamen Plotparameter einstellbar.

Plotparameter Speichern/Laden

Die in Kapitel 4.7.4 beschriebene automatische Parametersicherung in der aktuell bearbeiteten DDB-Datei wird beim Modulwechsel grundsätzlich immer durchgeführt. Darüberhinaus stehen im Dateimenu des **CAM-Prozessors** die Funktionen Laden Parameter und Speichern Parameter zur Verfügung. Mit Speichern Parameter können die CAM-Parameter wahlweise in einer anderen DDB-Datei gespeichert werden. Mit Laden Parameter können CAM-Parametereinstellungen aus einer selektierbaren DDB-Datei geladen und für das aktuell geladene Layoutelement aktiviert werden.

Plot/CAM Nullpunkt

Für die Ausgabe von Koordinaten muss ein Bezugspunkt definiert sein, der den Nullpunkt des Plotkoordinatensystems darstellt. Dieses Koordinatensystem besitzt nach rechts und oben hin steigende Werte. Der Nullpunkt wird beim Laden eines Elements in die linke untere Ecke der Elementabgrenzung gelegt. Mit Plot/CAM Nullpunkt kann der Bezugspunkt an eine beliebige Position gebracht werden. Dabei ist zu beachten, dass nicht alle Ausgabegeräte negative Koordinaten verarbeiten können. Bei der Defaultposition des Nullpunkts sind positive Koordinaten sichergestellt.

Im Plotparameterdialog der Windows- und Motifversionen des **CAM-Prozessors** besteht über einen entsprechenden Button zusätzlich die Möglichkeit, den Plotnullpunkt auf die Defaultposition zurückzusetzen.

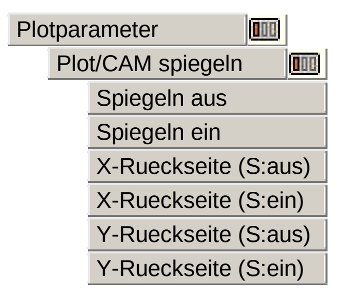
Plot/CAM Drehung

Mit Plot/CAM Drehen kann die Ausgabe um 90 Grad gegen den Uhrzeigersinn gedreht erfolgen. Mit 0 Grad wird wie auf dem Bildschirm dargestellt geplottet.

Zur Kennzeichnung des selektierten CAM-Rotationsmodus wird bei aktivierter Spiegelung bzw. Rotation am Marker für den CAM-Nullpunkt ein Beispieltext und ein Pfeil entsprechend des gewählten Spiegelungs- und Rotationsmodus angezeigt.

Plot/CAM Spiegelung

Um Texte z.B. auf Unterseiten von Platinen leserichtig zu plotten, bestehen verschiedene Möglichkeiten der Spiegelung. Diese sind in der Funktion Plot/CAM spiegeln zusammengefasst:



Spiegeln aus erzeugt eine Ausgabe wie am Bildschirm dargestellt. Spiegeln ein spiegelt alle Koordinaten an der durch den Nullpunkt verlaufenden X-Achse. X-Rueckseite (S:aus) spiegelt den Plot um die X-Achse, der Text bleibt jedoch leserichtig. X-Rueckseite (S:ein) spiegelt nur die Texte um ihre X-Achse. Mit Y-Ruecks (S:aus) wird um die X-Achse und der Text zusätzlich um seine Y-Achse gespiegelt. Y-Rueckseite (S:ein) plottet wie am Bildschirm dargestellt, die Texte werden um ihre Y-Achse gespiegelt. Abbildung 4-10 zeigt an einem Beispiel die Wirkungsweise der verschiedenen Spiegelfunktionen.

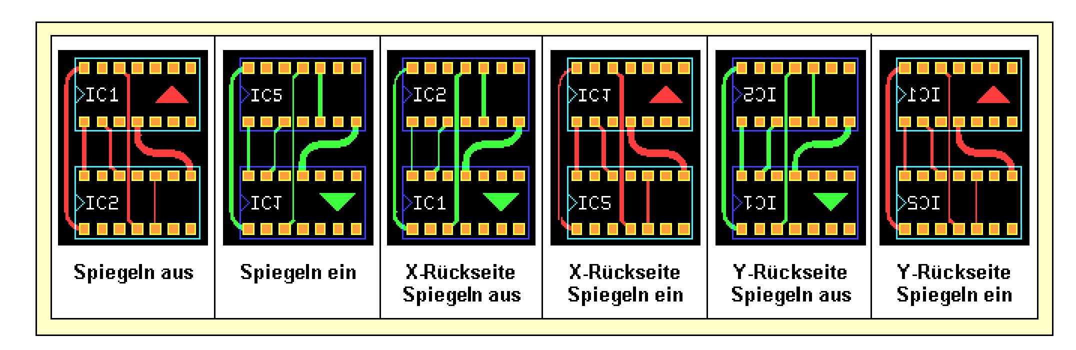


Abbildung 4-10: CAM-Spiegelungsarten

Bei allen Spiegelungen ist wieder auf die Lage des Nullpunkts zu achten. Bei Spiegeln ein muss er zum Beispiel in der linken oberen Elementbegrenzung liegen, um ausschließlich positive Koordinaten zu erzeugen. Befand sich der Plotnullpunkt an der Defaultposition, so springt er bei Wahl eines der Spiegelmenüpunkte automatisch in eine Position, die positive Koordinaten sicherstellt.

Zur Kennzeichnung des selektierten CAM-Spiegelungsmodus wird bei aktivierter Spiegelung bzw. Rotation am Marker für den CAM-Nullpunkt ein Beispieltext und ein Pfeil entsprechend des gewählten Spiegelungs- und Rotationsmodus angezeigt.

Alle Lagen Modus

Die Funktion Alle Lagen Modus gibt an, wie die Objekte zu behandeln sind, die auf den Signallagen Alle Lagen und Innenlagen bzw. auf beiden Seiten der Dokumentarlagen definiert sind. Bei Dokumentarlagen bezieht sich der Alle Lagen Modus auf Seite 1 und Seite 2 und entspricht der Auswahl Beide Seiten in den Dokumentarlagen-Menüs. Bei Signallagen bezieht sich der Alle Lagen Modus auf die Signallagenauswahl Alle Lagen bzw. Innenlagen. Mit der Option Getrennte Lage gibt man an, dass Objekte auf diesen Lagen nur bei Wahl der entsprechenden Plotlage (Alle Lagen, Innenlagen oder Beide Seiten bei Dokumentarlagen) ausgegeben werden. Mit Zusammenplotten werden diese Objekte zusammen mit der jeweils selektierten Signal- oder Dokumentarlage geplottet. D.h. mit Zusammenplotten werden beim Plotten der Signallage 1 auch die Objekte der Signallage Alle Lagen mit ausgegeben, beim Plotten der Seite 2 der Dokumentarlage Bestueckungsplan werden die Objekte der Dokumentarlage Bestueckungsplan - Beide Seiten mit ausgegeben, usw. Die Defaulteinstellung für den Alle Lagen Modus ist Zusammenplotten.

In der **BAE HighEnd**-Version werden für den Alle Lagen Modus zusätzlich die Optionen Angeschlossene, Pins & angeschlossene Vias und Vias & angeschlossene Pins angeboten. Diese Modi steuern die Ausgabe von Innenlagenpads für Pins und Vias. Mit Angeschlossene werden beim Plotten von Innenlagen nur die Pin- und Viaflächen, die auf der zu plottenden Innenlage Verbindung zu anderen Elementen haben, ausgegeben. Mit Pins & angeschlossene Vias werden beim Plotten von Innenlagen alle Pinflächen und nur die Viaflächen, die auf der zu plottenden Innenlage Verbindung zu anderen Elementen haben, ausgegeben. Mit Vias & angeschlossene Pins werden beim Plotten von Innenlagen alle Viaflächen und nur die Pinflächen, die auf der zu plottenden Innenlage Verbindung zu anderen Elementen haben, ausgegeben. Als Innenlagen werden dabei alle Signallagen mit Ausnahme der Signallage 1 und der über den Parameter Oberste Lage spezifizierten Lage betrachtet.

Symboltoleranz

Symboltoleranz gibt den maximalen Fehler vor, der beim Ausfüllen von Flächen mit den gegebenen Stiften bzw. Blenden gemacht werden darf. Dieser Fehler wird immer nur zum inneren Bereich der Fläche hin gemacht, d.h. es wird nicht über die vorgegeben Fläche hinaus gezeichnet (da ja sonst Kurzschlüsse erzeugt werden könnten). Der Defaultwert beträgt 0.15mm.

Ist eine gegebene Struktur auch mit der kleinsten zur Verfügung stehenden Blende bzw. dem Stift nicht zu realisieren, so verwendet der **CAM-Prozessor** die kleinste Blende, bzw. den zur Verfügung stehenden Stift und zeigt nach dem Plot Überzeichnungsfehler an. Ggf. falsch gezeichnete Strukturen werden durch Highlight gekennzeichnet. Mit Fehler loeschen im Menü Diverse lässt sich diese Fehleranzeige wieder zurücksetzen.

Umrandung und Passermarken

Mit den Funktionen Umrandung Modus und Passermarken Lage kann angegeben werden, ob die Umrandung bzw. die Passermarken zusammen mit allen anderen Plotlagen auszugeben sind (Optionen Umrandung plotten bzw. Passermarken ein) oder die Objekte auf der Umrandungs- bzw. Passermarkenlage nur bei Selektion der entsprechenden Plotlage ausgegeben werden (Optionen Keine Umrandung bzw. Passermarken aus). Die Defaulteinstellungen sind Umrandung plotten und Passermarken ein. Die Passermarkenlage ist die mit dem **BSETUP**-Kommando **`LAYPLTMARKLAY`** definierte Dokumentarlage für Film Passermarken (siehe hierzu auch Kapitel 7.1).

4.7.6 Versorgungslagen

Versorgungslagen werden vom **CAM-Prozessor** automatisch erzeugt. Dazu wird ein Negativbild der Versorgungslage generiert. Die auf den Versorgungslagen definierten Texte werden mit der aktuell eingestellten Standardlinien- bzw. Standardstiftbreite ausgegeben.

Während der Bearbeitung von Versorgungslagen sind dem **Layouteditor** die nachfolgend im **CAM-Prozessor** eingestellten Versorgungslagen-Plotparameter selbstverständlich noch nicht bekannt. Aus diesem Grund kann der Design Rule Check des **Layouteditors** auch keine umfassende Prüfung der später tatsächlich geplotteten Versorgungslagen durchführen. Wir empfehlen daher mit Nachdruck, die Versorgungslagen-Plots vor einer Weitergabe zum Filmhersteller oder zur Leiterkartenfertigung (z.B. mit **CAM-View**) einer eingehenden visuellen Kontrolle zu unterziehen; insbesondere sollte überprüft werden, ob z.B. durch zu groß gewählte Isolationsbreitenparameter ungewollt Inseln in Power Planes erzeugt wurden.

Versorgungslagen-Umrandung

Damit das Kupfer nicht bis zum Platinenrand reicht, wird entlang des Randes eine Isolationslinie gezogen, deren Breite sich mit dem Plotparameter V-Lagen-Umrandung einstellen lässt. Der Defaultwert hierfür beträgt 2.1mm. Enthält die Umrandung Kreisbögen, dann muss für die Plotausgabe ein Werkzeug (Gerber-Blende, Stift- bzw. Standardbreite) zur Verfügung stehen, mit dem die Umrandung in der angegebenen Breite gezogen werden kann.

Geteilte Potentiallagen und Potentialflächenisolation

Sofern Potentialflächen auf der zu plottenden Versorgungslage definiert sind, werden diese automatisch durch die Ausgabe der entsprechenden Flächenumrandung als Isolationslinie generiert. Die Bearbeitung aktiver Potentialflächen mit Netzzuweisung führt dabei zur Generierung sogenannter geteilter Potentiallagen, während über passive Potentialflächen ohne Netzzuweisung echte Isolationsflächen in der Versorgungslage erzeugt werden. Die Breite der Potentialflächenisolation kann mit der Funktion V-Lagen-Isolation definiert werden. Der hierfür voreingestellte Defaultwert beträgt 0.3mm.

Fremdnetzisolation

Um die nicht zum Netz der Versorgungslage bzw. Potentialfläche gehörenden Pins abzuisolieren, werden um die Bohrungen dieser Pins ausgefüllte Kreise gezeichnet. Die Durchmesser dieser Kreise werden durch die Bohrdurchmesser und die Parameter V-IS-Min.Abstand (Mindestabstand Isolation zu Bohrung) und V-IS-Toleranz (Toleranz Abstand Isolation zu Bohrung) bestimmt. Der Durchmesser eines solchen Isolationskreises beträgt minimal

```
Bohrdurchmesser + 2 × (V-IS-Min.Abstand)
```

und maximal

```
Bohrdurchmesser + 2 × (V-IS-Min.Abstand) + 2 × (V-IS-Toleranz)
```

Abbildung 4-11 verdeutlicht die Wirkungsweise der Parametereinstellungen für den Mindestabstand und die Toleranz der Versorgungslagen-Isolation.

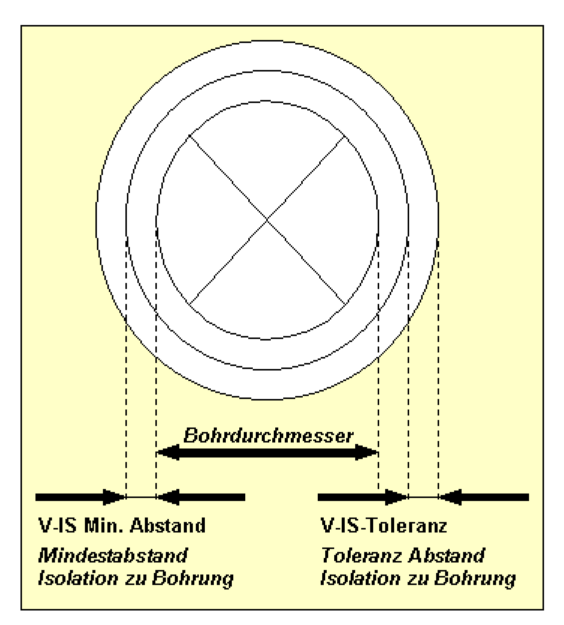


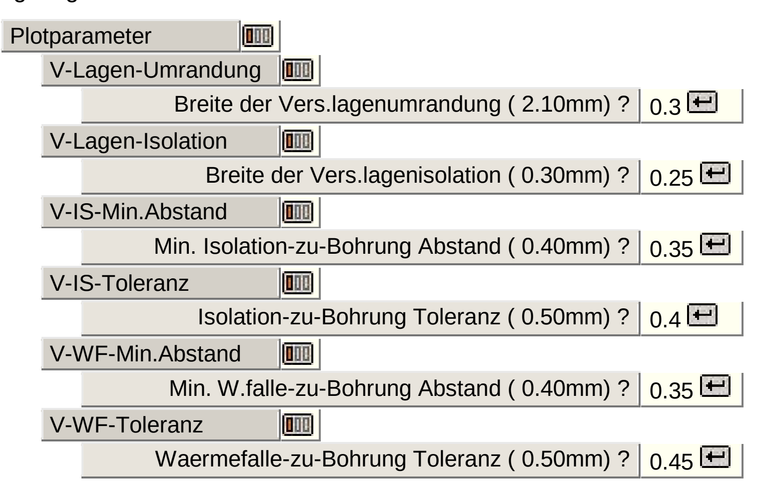
Abbildung 4-11: CAM-Versorgungslagenisolation

Wärmefallen

Um die zum Netz der Versorgungslage bzw. Potentialfläche gehörenden Pins werden automatisch Wärmefallen gezeichnet. Diese bestehen aus vier Achtelkreislinien, die um die Bohrung herum angeordnet sind, um zu verhindern, dass beim Löten zu viel Wärme abfließen kann. Die Durchmesser dieser Kreise werden analog zu den Isolationskreisen durch die Parameter V-WF-Min.Abstand (Mindestabstand Wärmefalle zu Bohrung) und V-WF-Toleranz (Toleranz Abstand Wärmefalle zu Bohrung) bestimmt. Die Defaultwerte betragen für die Mindestabstände 0.4mm und für die Toleranzen 0.5mm.

Beim Fotoplot wird vor dem Zeichnen der Wärmefalle noch geprüft, ob eine Blende vom Typ thermisch definiert ist, deren Durchmesser im berechneten Toleranzbereich liegt. Ist eine entsprechende Blende vorhanden, so wird die Wärmefalle mit dieser geblitzt. Ansonsten werden die Wärmefallen mit der kleinsten runden Blende mit Modus all oder line gezogen.

Nehmen Sie mit den folgenden Kommandos eine Parametereinstellung für Versorgungslagen auf einen Mindestisolationsabstand von 0.35mm mit einer Toleranz von 0.4mm, d.h. maximal 0.75mm Isolationsabstand vor; die Wärmefallen sollen in einem Abstand von mindestens 0.35mm mit einer Toleranz von 0.45mm gezogen werden, d.h. in einem Abstand von maximal 0.8mm; die Versorgungslagenumrandung soll mit 0.3mm, die Potentialflächen-Isolation mit 0.25mm gezogen werden:



4.7.7 HP-GL-Ausgabe

Die Erzeugung von HP-GL-Plots geschieht mit der Funktion Plot HP-GL Ausgabe im Menü Kontrollplot.

Die Ausgabe erfolgt dabei mit einem Stift, dessen Breite über die Funktion Stift-/Standardbreite angeben werden kann. Von der Breite dieses Stiftes hängt es ab, bis zu welcher minimalen Größe Strukturen ohne Überzeichnungsfehler ausgegeben werden können. Die Defaultstiftbreite beträgt 0.3mm. Es können Werte im Bereich von 0.01 bis 10.0mm angegeben werden.

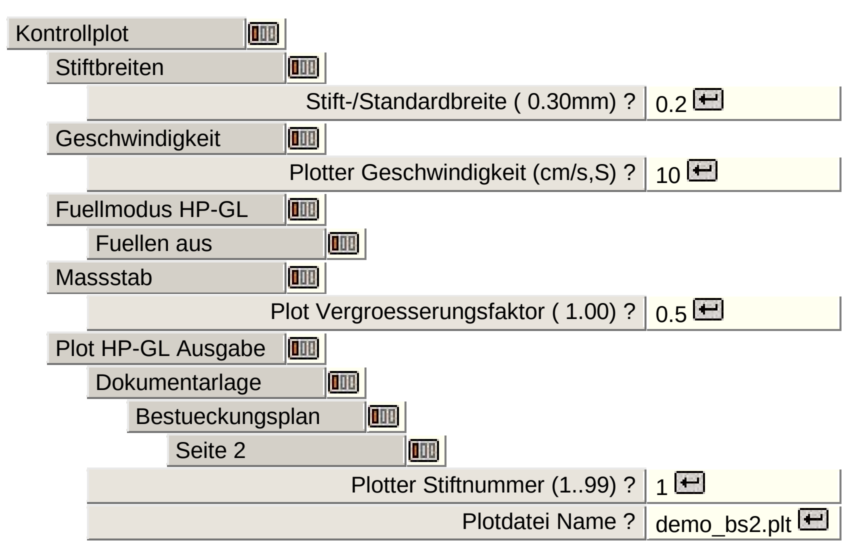
Die Plotgeschwindigkeit kann mit Geschwindigkeit beeinflusst werden. Dabei kann man die Geschwindigkeit in cm/s oder S für maximale Geschwindigkeit angeben. Die Defaulteinstellung ist **S**. Als maximaler Zahlenwert für die Geschwindigkeit wird 99 akzeptiert.

Die Koordinaten werden in Plotter Units ausgegeben. Ein Millimeter entspricht 40.2 Plotter Units. Einige "HP-GL-kompatible" Plotter legen 40 Plotter Units pro Millimeter für die Ausgabe zugrunde. Der dabei auftretende Fehler kann durch Wahl eines entsprechenden Faktors mit Hilfe der Funktion Massstab kompensiert werden. Der Defaultwert für den Maßstab beträgt 1.0. Es können Werte von 0.1 bis 100.0 angegeben werden.

Für schnelle Kontrollplots kann man im Menü Fuellmodus HP-GL mit Fuellen aus das Ausfüllen von Flächen unterbinden. Es wird dann bei Flächen nur die Umrandungslinie geplottet. Mit Fuellen ein kann der Defaultzustand wieder hergestellt werden.

Die HP-GL-Plotausgabe wird mit der Funktion Plot HP-GL Ausgabe gestartet. Der Benutzer kann dabei zunächst eine Lage aus dem üblichen Lagenauswahlmenü wählen und wird dann um die Angabe einer Stiftnummer (1..99) gebeten.

Erstellen Sie mit den folgenden Kommandos einen HP-GL-Plot (Datei **demo_bs2.plt**) der Seite 2 der Dokumentarlage Bestueckungsplan mit nicht ausgefüllten Flächen in halber Größe mit Stift 1, einer Stiftbreite von 0.2mm und einer Plotgeschwindigkeit von 10 cm/s:



Beachten Sie die Option Mehrere Lagen im Lagenauswahlmenü der Funktion Plot HP-GL Ausgabe. Damit können die Plotdaten *mehrerer*, selektierbarer Lagen simultan auf eine einzige Datei ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Plotlagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Die Lagenselektion mit der linken Maustaste erlaubt hierbei zusätzlich die Spezifikation lagenspezifischer Stiftnummern (per Default wird für jede Lage der Stift 1 verwendet). Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren; dabei werden die Stiftnummern für die HP-GL-Ausgabe automatisch auf die Modulo-8-Werte der entsprechenden Lagenfarben gesetzt.

Das erfolgreiche Schreiben des HP-GL-Plots wird durch die Meldung

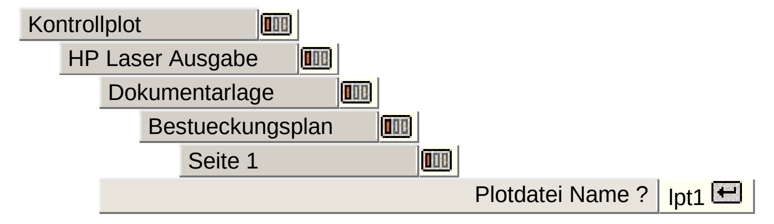
```
HP-GL Plot beendet (<n> Fehler).
```

quittiert. Dabei gibt **<n>** die Anzahl der beim Plot gemachten Überzeichnungsfehler an. Die zu groß gezeichneten Flächen werden in Highlight-Farbe dargestellt.

Am Ende der HPGL-Ausgabedateien wird ein **PG**-Kommando zum Auswurf der Seite abgesetzt, um zu vermeiden, dass aufeinanderfolgende Blattausgaben übereinander geplottet werden. Ist die Ausgabe mehrerer Lagen auf ein Blatt gewünscht, so ist bei der Lagenauswahl die Option Mehrere Lagen zu verwenden.

4.7.8 HP-Laser-Ausgabe

Die Ausgabe von HP-Laser-Plots im Format PCL (Printer Command Language) kann mit der Funktion HP Laser Ausgabe vom Menü Kontrollplot aus gestartet werden. Der Plot wird dabei automatisch auf A4 skaliert, d.h. weder die Einstellung des Maßstabs, noch die Angabe einer Stift- bzw. Standardbreite haben hier Wirkung. Um die Dokumentarlage Bestueckungsplan (Seite 1, Lötseite) des aktuell geladenen Layouts auf **lpt1** (d.h. hier z.B. Direktangabe der DOS-Schnittstelle zu Laserdrucker) auszugeben, sind folgende Kommandos auszuführen:



Beachten Sie die Option Mehrere Lagen im Lagenauswahlmenü der Funktion HP Laser Ausgabe. Damit können die Plotdaten *mehrerer*, selektierbarer Lagen simultan auf eine einzige Datei ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Plotlagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren.

Das erfolgreiche Schreiben des PCL-Plots wird durch die Meldung

```
HP Laser Ausgabe beendet (Skalierung 1:...).
```

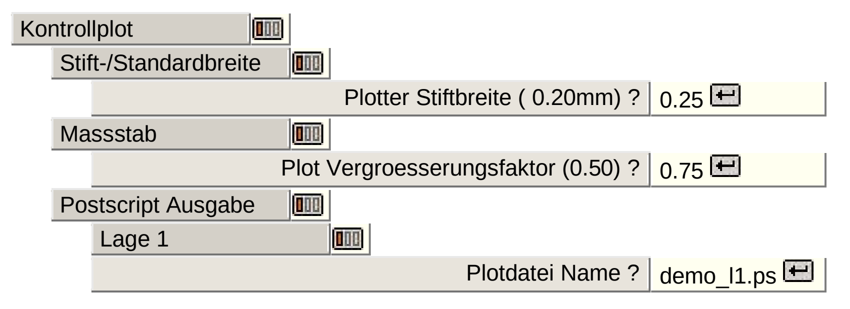
mit der Angabe des zur Skalierung auf Blattgröße verwendeten Skalierungsfaktors quittiert.

Die Daten müssen im Binärmodus an den Ausgabekanal übertragen werden. Erfolgt die Ausgabe zunächst auf eine externe Datei (z.B. **demo_bs1.pcl**), dann ist auf DOS-Ebene bei der anschließenden Übertragung dieser Datei an den Laserdrucker mit Hilfe des COPY-Befehls die Option **/b** anzuwenden:

```
>  copy demo_bs1.pcl lpt1 /b
```

4.7.9 Postscript-Ausgabe

Die Postscript-Ausgabe wird mit der Funktion Postscript Ausgabe im Menü Kontrollplot gestartet. Um die Signallage 1 des aktuell geladenen Layouts mit einem Vergrößerungsfaktor von 0.75 und unter Verwendung einer Standardlinienbreite von 0.25 mm auf die Datei `demo_l1.ps` auszugeben, sind folgende Kommandos auszuführen:



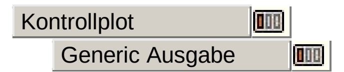
Beachten Sie die Option Mehrere Lagen im Lagenauswahlmenü der Funktion Postscript Ausgabe. Damit können die Postscriptdaten *mehrerer*, selektierbarer Lagen simultan auf eine einzige Datei ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Plotlagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren.

Das erfolgreiche Schreiben der Postscript-Datei wird durch die Meldung **`Postscript Ausgabe beendet.`** quittiert.

4.7.10 Generische Ausgabe unter Windows

In den Windows-Versionen der BAE-PC-Software ist eine generische Plot- bzw. Druckerausgabe implementiert. Damit werden durch den **CAM-Prozessor** der BAE-Windows-Software prinzipiell *alle* unter der aktuell definierten Windows-Betriebssystemkonfiguration verfügbaren Print- bzw. Plotfunktionen unterstützt.

Zur Aktivierung des Windows Print-/Plot-Menü sind die folgenden Kommandos auszuführen:



Beachten Sie die Option Mehrere Lagen im Lagenauswahlmenü der Funktion Generic Ausgabe. Damit können die Plotdaten *mehrerer*, selektierbarer Lagen simultan auf eine einzige Datei ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Plotlagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Die Lagenselektion mit der linken Maustaste erlaubt hierbei über die Zuweisung von Stiftnummern zur Angabe von Indizes in die aktuell definierte Farbtabelle zusätzlich die Spezifikation lagenspezifischer Farben für eine *farbige* Plotausgabe. Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren; dabei werden die Farben für die Plotausgabe automatisch auf die Modulo-8-Werte der lagenspezifischen Farbtabellenindizes gesetzt.

Die im Windows-Druckerdialog vorgenommenen Einstellungen für die Anzahl der Kopien, die Sortierung sowie den Seitenausgabebereich werden bei der generischen Ausgabe berücksichtigt.

Bei Anwahl der Option Alle Seiten im Druckerdialog der generischen Ausgabe unter Windows werden alle Seiten ausgegeben. Somit ist es möglich z.B. alle Layouts eines Projektes auf einmal auszudrucken. Um z.B. Layouts gemischt gedreht und nicht gedreht ausgeben zu können, werden jeweils die für das zu plottende Element eingestellten Druckparameter berücksichtigt. Diese können sich von den Parametern des aktuell geladenen Elements unterscheiden.

Bei Anwahl der Option Markierung im Druckerdialog der generischen Ausgabe unter Windows kann ein Bereich für die Plotausgabe selektiert werden.

Bei der generischen Ausgabe wird eine automatische Anpassung der Skalierung auf das für die Druckausgabe definierte Blattformat unter Beibehaltung des Seitenverhältnisses vorgenommen, wenn die Größe des zu plottenden Elements die über die Druckereinstellung definierte Blattgröße überschreitet. Die Funktion Generic Ausgabe zeigt nach Beendigung der Ausgabe in der Mitteilungszeile den zur Skalierung auf Blattgröße verwendeten Skalierungsfaktor an.

4.7.11 Bitmap-Plotausgabe auf die Windows-Zwischenablage

In der Windowsversion ermöglicht die Funktion Ausgabe nach Zwischenablage aus dem Menü Kontrollplot die Ausgabe von Zeichnungsdaten in eine Bitmap, die in die Zwischenablage zum weiteren Verarbeiten durch (Einfügen) in andere bitmapfähige Windowsanwendungen übertragen wird. Per Default wird das gesamte aktuell geladene Element ausgegeben. Mit Clipping ein lässt sich die Ausgabe auf ein mausselektierbares Rechteck beschränken. Die Dialogbox zur Plotparametereinstellung erlaubt auch eine Größenvorgabe für die Bitmap, sowie die Auswahl des Rotations- und Spiegelungsmodus für die Ausgabe.

Bei Auswahl einer einzelnen Ausgabelage erfolgt die Ausgabe schwarz auf weißem Grund. Bei Wahl von Mehrere Lagen erfolgt die Ausgabe der selektierten Lagen entsprechend den Einstellungen der Farbpalette in Mischfarben auf schwarzem Hintergrund.

4.7.12 Gerber-Photplot

Für das Erstellen von Gerber-Plots benötigt der **CAM-Prozessor** eine Tabelle mit den zur Verfügung stehenden Blenden. Zur Erstellung und Verwaltung solcher Blendentabellen bietet das Menü Gerber Fotoplot einige Grundfunktionen an. Die Blendentabellen werden dabei global in der Datei **`cam.dat`** gehalten. Erstellt man eigene Blendentabellen, so sollte man darauf achten, dass man bei System-Updates diese Datei nicht überspielt (siehe hierzu auch die Bartels AutoEngineer® Installationsanleitung).

Gerber Konstruktionstechniken

Es empfiehlt sich, möglichst effiziente Gerber-Blendentabellen zu definieren, um die Datenmengen bei der Plotdatenausgabe zu minimieren. Der Grund hierfür liegt darin, dass unnötig große Gerberdateien mehr Speicherplatz benötigen und damit längere Kopier- und Modem-Übertragungszeiten beanspruchen. Sofern die Plots auf einem Vektorplotter erzeugt werden, ergeben sich bei größeren Plotdateien auch längere Plotzeiten und damit höhere Produktionskosten. Die Fotoplotter unterstützen zwei verschiedene Zeichenmodi für die Ausgabe liniengezogener und geblitzter Strukturen. Eine geblitzte Struktur wird auf dem Plotfilm durch eine einzige Blitzbelichtung mit der entsprechend selektierten Blende erzeugt. Flächen, für die in der Blendentabelle eine passende Blitzbelichtungs-Blende existiert, lassen sich demnach durch ein Flash-Kommando mit einem einzigen Koordinatenpaar realisieren. Flächen ohne passende Blende werden soweit möglich mit Hilfe effizienter Zeichnungstechniken (siehe unten) realisiert. Sofern es sich hierbei um einfach nachzubildende Strukturen handelt, lässt sich die Anzahl der Ausgabekoordinaten in Grenzen halten. Für unregelmäßig geformte bzw. platzierte Strukturen sowie für Flächen ohne passende Blenden hingegen kommen Gerber-Füllverfahren zur Anwendung, die u.U. erhebliche Datenmengen produzieren. Mit einer geeigneten Definition der Padformen und der Verwendung passender Blendentabellen können demnach der Speicherplatzbedarf für generierte Gerberdateien, der Zeitaufwand für Kopiervorgänge, die Kosten für Modemübertragungen, usw. entscheidend minimiert werden.

Orthogonal platzierte quadratische bzw. rechteckige Pads mit passender quadratischer bzw. rechteckiger Blende werden geblitzt. Orthogonal platzierte rechteckige Pads, für die eine quadratischen Blende mit einer Kantenlänge entsprechend der kürzeren Padseite existiert, werden durch einen Linienzug realisiert. Orthogonal platzierte rechteckige (und quadratische) Pads, für die eine quadratische Blende mit einer Kantenlänge größer als die Hälfte der kürzeren Padseite existiert, werden durch zwei Linienzüge mit der entsprechenden Blende realisiert. Nicht im rechten Winkel platzierte rechteckige Pads werden über Füllverfahren realisiert. Kreisförmige Pads mit einer passenden runden Flash-Blende werden geblitzt. Kreisförmige Pads ohne passende Blende werden durch kreisförmige Linienzüge mit der nächstkleineren runden Blende realisiert. Fingerförmige Pads (Rechtecke mit Halbkreisen an den Enden) werden wie Leiterbahnsegmente behandelt, d.h. sie werden ggf. mit einer entsprechenden runden Blende als einfache Linie abgefahren. Pads, die den vorgenannten Spezifikationen nicht genügen, werden über Füllverfahren realisiert. Leiterbahnzüge mit einer Breite entsprechend einer verfügbaren runden Line-Blende werden durch Linienzüge realisiert. Leiterbahnzüge, für deren Breite keine Blende zur Verfügung steht, werden durch sich überlappende Linienzüge mit der nächstkleineren Line-Blende realisiert. Unregelmäßig geformte bzw. platzierte Flächen werden grundsätzlich über Füllverfahren realisiert.

Gerber Blendentabelle

Die Funktionen Blenden laden, Blenden speichern und Blenden loeschen führen die entsprechenden Grundfunktionen mit den Blendentabellen durch. B-Tab.Verzeichnis zeigt die Namen aller ladbaren Blendentabellen an.

Tritt beim Laden die Meldung **`Gerber Blendentabelle Ueberlauf!`** auf, so bedeutet dies, dass in der geladenen Blendentabelle mehr als die vom System verwaltbaren 900 Einträge enthalten sind. Da man solche Blendentabellen nicht innerhalb des **Bartels AutoEngineer** erzeugen kann, deutet diese Meldung auf eine korrupte **`cam.dat`**-Datei hin.

Bei Starten des **CAM-Prozessors** wird automatisch die Blendentabelle mit dem Namen **`standard`** geladen. Die darin definierten Blenden sind in Tabelle 4-4 aufgelistet.

Tabelle 4-4: Gerber Blendentabelle "standard"

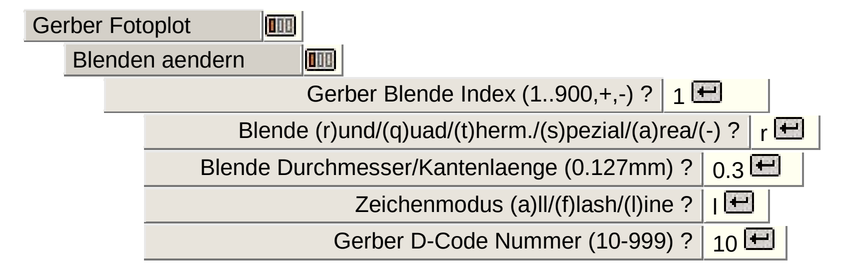
		Blendengröße		
D-Code	**Blendentyp**	**[mil]**	**[mm]**	**Zeichenmodus**
D10	rund	7.87	0.200	ALL
D11	rund	8.27	0.210	ALL
D12	rund	9.84	0.250	ALL
D13	rund	11.81	0.300	ALL
D14	rund	15.75	0.400	ALL
D15	rund	19.69	0.500	ALL
D16	rund	23.62	0.600	ALL
D17	rund	27.56	0.700	ALL
D18	rund	31.50	0.800	ALL
D19	rund	35.43	0.900	ALL
D20	rund	39.37	1.000	ALL
D21	rund	43.31	1.100	ALL
D22	rund	47.24	1.200	ALL
D23	rund	51.18	1.300	ALL
D24	rund	59.06	1.500	ALL
D25	rund	62.99	1.600	ALL
D26	rund	66.93	1.700	ALL
D27	rund	78.74	2.000	ALL
D28	rund	90.55	2.300	ALL
D29	rund	98.43	2.500	ALL
D30	rund	102.36	2.600	ALL
D31	rund	110.24	2.800	ALL
D32	rund	118.11	3.000	ALL
D33	rund	129.92	3.300	ALL
D34	rund	137.80	3.500	ALL
D35	rund	149.61	3.800	ALL
D36	rund	157.48	4.000	ALL
D37	rund	169.29	4.300	ALL
D38	quadratisch	15.75	0.400	ALL
D39	quadratisch	19.69	0.500	ALL
D40	quadratisch	23.62	0.600	ALL

D-Code	Blendentyp	Blendengröße [mil]	[mm]	Zeichenmodus
D41	quadratisch	29.53	0.750	ALL
D42	quadratisch	31.50	0.800	ALL
D43	quadratisch	39.37	1.000	ALL
D44	quadratisch	43.31	1.100	ALL
D45	quadratisch	47.24	1.200	ALL
D46	quadratisch	51.18	1.300	ALL
D47	quadratisch	59.06	1.500	ALL
D48	quadratisch	62.99	1.600	ALL
D49	quadratisch	78.74	2.000	ALL
D50	quadratisch	86.61	2.200	ALL
D51	quadratisch	118.11	3.000	ALL
D52	quadratisch	129.92	3.300	ALL
D53	thermisch	70.87	1.800	ALL
D54	thermisch	86.61	2.200	ALL
D55	thermisch	98.43	2.500	ALL

Das Editieren von Blendentabellen erfolgt mit der Funktion Blenden aendern. Nach Aktivieren dieser Funktion erscheint am Bildschirm eine Übersicht über die ersten Einträge der Blendentabelle. Man kann nun durch Eingabe von + bzw. - jeweils eine Übersichtsseite vor- bzw. zurückblättern. Durch Betätigen der Eingabetaste ↵ gelangt man zurück zur Menüverwaltung. Durch Eingabe eines Index kann man den entsprechenden Eintrag der Blendentabelle ändern. Für die Parameter gilt bei Betätigen der Eingabetaste ↵ jeweils der alte Wert. Zunächst wird die Form mit Hilfe eines Kürzels angegeben. `r` steht für rund, `q` für quadratisch, `a` für rechteckig ("Area"), `t` für thermisch und `s` für spezial, wobei die Spezialeinträge vom System nicht verwendet werden. Danach gibt man den Durchmesser bzw. die Kantenlänge(n) der Blende an. Darauf folgt die Eingabe des Modus. `f` (Flash) bedeutet, dass mit diesem D-Code nur geblitzt werden kann. `l` (Line) steht für die alleinige Verwendung als Line-Blende und `a` (All) für die Verwendbarkeit als Line- und Flash-Blende. Zuletzt wird noch der D-Code der Blende angegeben. Es ist darauf zu achten, dass keine D-Codes mehrfach eingetragen werden, da das System dabei je nach Bedarf die eine oder andere Definition verwendet. Gibt man auf die Abfrage nach der Form `-` an, so wird der entsprechende Eintrag in der Blendentabelle freigegeben.

Beachten Sie, dass quadratische und rechteckige Blenden nur zum Blitzen orthogonal platzierter Strukturen verwendet werden können. Vor der Verwendung rechteckiger Blenden ist außerdem unbedingt zu klären, ob der Leiterkartenhersteller diesen Blendentyp verarbeiten kann.

Tragen Sie unter dem D-Code 10 mit den folgenden Kommandos eine runde Line-Blende mit 0.3mm Durchmesser in den ersten Blendentabellenplatz ein:



Durch Betätigen der Eingabetaste ↵ gelangen Sie wieder in die Menüoberfläche. Mit Blenden ausgeben kann man eine ASCII-Ausgabe der geladenen Blendentabelle erzeugen. Mit Blenden speichern kann die aktuell definierte Blendentabelle unter einem frei wählbaren Namen (in der Datei `cam.dat`) gespeichert werden.

Gerber Format, Optimierte Gerberausgabe

Für die Ausgabe kann mit Gerber Format und Gerber 2.3 Format das Format mit ganzzahligen 1/1000 Inch Koordinaten, d.h. 3 Nachkommastellen, oder mit Gerber Format und Gerber 2.4 Format Ausgabe in 1/10000 Inch, d.h. 4 Nachkommastellen, gewählt werden. Defaulteinstellung ist Gerber 2.3 Format. Zusätzlich stehen die Optionen Gerber optimiert 2.3 und Gerber optimiert 2.4 zur Auswahl. Bei der optimierten Gerberausgabe werden gleichbleibende Koordinatenkomponenten und die Wiederholung des Kommandos D01 ("Licht aus") bei längeren Linienzügen unterdrückt. Dadurch ergibt sich eine signifikante Reduzierung der Plotdatenmengen. Vor der Anwendung der optimierten Gerberausgabe ist jedoch unbedingt zu klären, ob der Leitenplattenhersteller diese Optimierungen unterstützt.

Standard-Linienbreite

Die Linienbreite für Texte und Grafiklinien wird mit Std. Linienbreite eingestellt. Der Defaultwert beträgt 0.3mm. Diese Linienbreite muss innerhalb der gegebenen Toleranz mit einer runden Line-Blende realisierbar sein. Ansonsten gibt das System die Fehlermeldung **`Standard Linienbreite nicht realisierbar!`** aus.

Gerber Füllmodus

Über Gerber Fuellmodus kann das Verfahren spezifiziert werden, nach dem die Flächenstrukturen gezeichnet werden sollen. Hierbei stehen das Multiblenden-Füllverfahren (Defaulteinstellung Multiblendenfuellen; zum Flächeninneren hin immer größere Blenden) und das Linien-Füllverfahren (Linienfuellen; Füllen mit der kleinsten zur Verfügung stehenden Line-Blende) zur Auswahl.

Als weitere Option wird G36/G37 Fuellen angeboten. In diesem Modus werden die nicht blitzbaren Strukturen nicht durch Verfahren mit Blenden aufgefüllt, sondern direkt als Außenkontur der Füllfläche abgespeichert, und erst später durch den Photoplotter generiert. Dies führt zu erheblich reduzierten Datenmengen bei der Ausgabe von nicht blitzbaren Flächen. Ein weiterer unschätzbarer Vorteil dieses Verfahrens besteht darin, dass Überzeichnungsfehler praktisch ausgeschlossen werden können. Allerdings ist vor Verwendung des G36/G37-Füllmodus unbedingt mit dem Leiterplattenhersteller zu klären, ob der eingesetzte Photoplotter dieses Verfahren unterstützt.

Gerber Kreisbogenmodus

Mit der Funktion Gerber Kreisboegen kann wahlweise die Verwendung von Gerber-I/J-Kreisbefehlen (Option Gerberkreisbefehle) anstelle der Interpolation (Defaultoption Kreisinterpolation) aktiviert werden. Sofern der Fotoplotter Gerber-Kreisbefehle interpretieren kann, sollten diese auch verwendet werden, da sich dadurch eine erhebliche Datenreduktion in der Gerberausgabe erzielen lässt.

Extended Gerber

Über die Funktion Extended Gerber kann wahlweise eine Fotoplotausgabe im Format RS-274-X (Extended Gerber with Embedded Apertures) erfolgen. Es stehen die Optionen Kein Extended Gerber, Extended Gerber fest und Extended Gerber dynamisch zur Auswahl.

Kein Extended Gerber entspricht der Standardausgabe. Im Gegensatz dazu wird bei Extended Gerber in der Gerberdatei noch zusätzlich die verwendete Blendentabelle und das Gerberformat mit abgespeichert (d.h. in die Plotdatei eingebettet). Dabei kann zwischen fester und dynamisch (d.h. automatisch) generierter Blendentabelle gewählt werden. Die dynamisch generierte Blendentabelle bezieht sich auf das gesamte Layout, d.h. nicht nur die aktuell geplottete Lage. Der Vorteil der Extended Gerber Ausgabe besteht darin, dass die Blendentabelleninformation jeweils automatisch mit der Plotdatei an den Filmhersteller übermittelt werden kann. Bei Verwendung der dynamisch generierten Blendentabelle erübrigt sich darüber hinaus auch noch die sonst notwendige Definition bzw. Aktivierung einer Blendentabelle vor der Gerberplotausgabe.

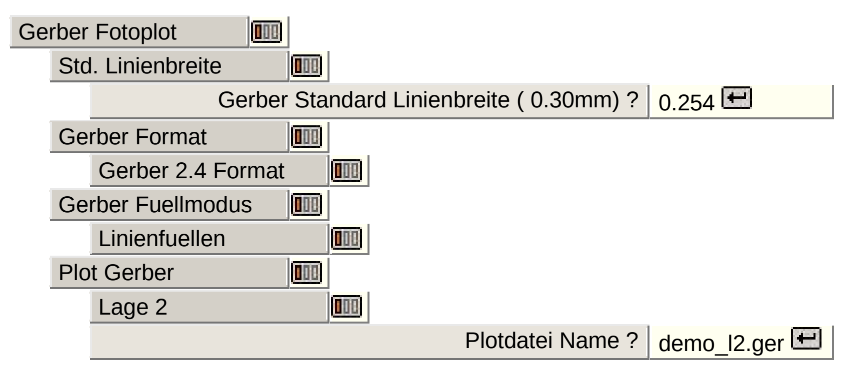
Warnung

Vor einem produktiven Einsatz des Extended Gerber-Formats ist unbedingt zu klären, ob der Leiterkartenhersteller diese Option unterstützt, d.h. ob der zur Filmerstellung verwendete Gerberplotter das Format Extended Gerber verarbeiten kann. Bei Anwendung des Multiblendenfüllverfahrens wird von der Verwendung automatisch generierter Blendentabellen ohnehin abgeraten, da sich die dynamisch generierten Blendentabellen nur in Ausnahmefällen für dieses Füllverfahren eignen.

Gerber Photoplot-Ausgabe

Mit Plot Gerber wird die Ausgabe von Gerberdaten gestartet. Dabei wird noch ein Auswahlmenü für die zu plottende Lage durchlaufen. Beachten Sie hierbei die Option Mehrere Lagen im Lagenauswahlmenü. Damit können die Photoplotdaten *mehrerer*, selektierbarer Lagen simultan ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Plotlagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren.

Geben Sie für die Signallage 2 des aktuell geladenen Layouts die Gerberdaten im Format 2.4 mit einer Standardlinienbreite von 0.254mm unter Verwendung der aktuell definierten Blendentabelle und im Linien-Füllmodus auf die Datei **`demo_l2.ger`** aus:



Dem Schreiben der Ausgabedatei folgt die Ausgabe eines Reports. Der Report enthält folgende Daten über die Ausgabe:

```
Anzahl blitzbelichteter Strukturen .....: <f>
Anzahl rechteck-gezeichneter Flaechen ..: <r>
Anzahl kreis-gezeichneter Flaechen .....: <c>
Anzahl Multiblenden Flaechen ...........: <m>
Anzahl liniengefuellter Flaechen .......: <l>
Anzahl gezeichneter Waermefallen .......: <w>
Anzahl Ueberzeichnungs-Fehler ..........: <e>
```

<f> gibt dabei die Anzahl der Strukturen an, die direkt durch Blitzen mit einer Blende erzeugt wurden. **<r>** ist die Zahl der rechteckigen Flächen, die mit quadratischer Line-Blende gefüllt wurden, **<c>** die der Kreise, die durch eine kleinere im Kreis gezogene runde Line-Blende erzeugt wurden. **<m>** steht für die Anzahl der Strukturen, die im Multiblenden-Füllmodus gezeichnet wurden. **<l>** steht für die Anzahl der Strukturen, die mit der kleinsten verfügbaren runden Line-Blende gefüllt wurden. **<w>** ist die Zahl der Wärmefallen, die mit der kleinsten verfügbaren runden Line-Blende gezeichnet wurden. **<e>** gibt die Anzahl der Strukturen an, die mit der kleinsten verfügbaren runden Line-Blende nicht innerhalb der gegebenen Toleranz erzeugt werden konnten.

In keinem Fall sollten Sie eine Fotoplot-Ausgabe, bei der Überzeichnungsfehler auftraten, in die Fertigung bzw. zum Filmhersteller geben. In der Regel führt die Weiterverarbeitung derartiger Plots zur Erzeugung von fehlerhaften Leiterkarten!

Mit Hilfe des **CAM-View**-Moduls (siehe Kapitel 4.8) können die erzeugten Gerberdaten visuell auf deren Richtigkeit überprüft werden.

4.7.13 Bohrdaten

Das Menü Bohr+Bestueckdaten enthält Funktionen zur Generierung von Bohrdaten in den Formaten Excellon II bzw. Sieb&Meier. Mit Hilfe des **CAM-View**-Moduls (siehe Kapitel 4.8) lassen sich die erzeugten Bohrdaten anschließend visualisieren, sortieren oder zur Nutzengenerierung weiterverarbeiten.

Excellon

Mit der Funktion Excellon Bohrdaten können Bohrdaten im Format Excellon II erzeugt werden. Hierbei werden sowohl die Bohrdaten selbst als auch die Werkzeug- bzw. Bohrertabelle in einer einzigen Datei ausgegeben.

Der Name der Excellon-Ausgabedatei kann mit Bandgeraet Kanal voreingestellt werden. Mit dem Parameter Werkzeugtoleranz kann die Toleranz angegeben werden, mit der bei der Auswahl von Bohrwerkzeugen vorgegangen werden soll. Der Defaultwert für die Bohrertoleranz beträgt 0.10mm.

In der Funktion Excellon Bohrdaten erfolgt vor der eigentlichen Bohrdatenausgabe die Abfrage nach der Bohrungsklasse, über die die selektive Ausgabe durchkontaktierter, nicht durchkontaktierter, oder zu partiellen Durchkontaktierungen gehörender Bohrungen möglich ist. Durch die Eingabe eines Bindestrichs **-** auf die Abfrage nach der Bohrklasse werden diejenigen Bohrungen für die Ausgabe selektiert, die keiner speziellen Bohrklasse zugeordnet sind. Die erfolgreiche Ausgabe der Bohrdaten wird mit der Meldung **`Bohrdaten geschrieben.`** quittiert.

Sieb & Meier

Mit den Funktionen Bohrband erstellen und Werkzeugtabelle erstellen können Bohrdaten und Werkzeugtabellen im Format Sieb&Meier erzeugt werden.

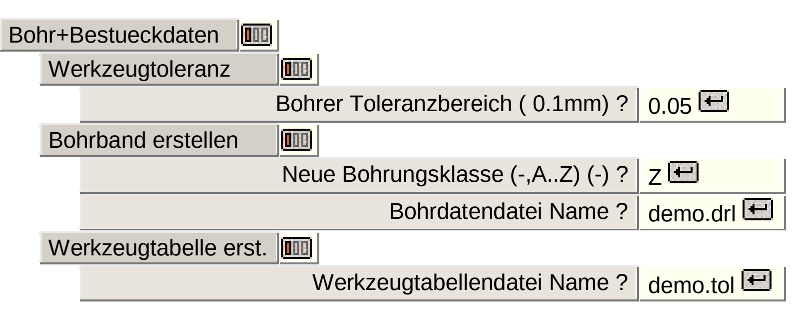
Die Namen der Ausgabedateien für die Sieb&Meier-Bohrdatenausgabe können mit Bandgeraet Kanal bzw. Tabellen-Drucker Kanal voreingestellt werden. Mit dem Parameter Werkzeugtoleranz kann die Toleranz angegeben werden, mit der bei der Auswahl von Bohrern vorgegangen werden soll. Der Defaultwert für die Bohrertoleranz beträgt 0.10mm.

Die erzeugte Werkzeugtabelle enthält eine Auflistung aller verwendeten Bohrer mit ihren zugeordneten Bohrernummern. Diese Tabelle kann bis zu 99 Einträge enthalten. Sollten mehr verschiedene Bohrer notwendig sein, bricht die Bohrdatenausgabe mit Meldung **`Es sind zu viele Bohrdurchmesser definiert!`** ab.

Im Sieb&Meier-Format werden für jede Bohrung die Bohrkoordinaten in der Einheit 1/100mm in einer eigenen Zeile ausgegeben. Bei der ersten Verwendung eines neuen Bohrdurchmessers wird dessen Bohrernummer am Ende der Zeile angeführt.

Mit Bohrband erstellen wird die Ausgabe der Bohrdaten gestartet, Werkzeugtabelle erstellen bewirkt die Ausgabe der Werkzeugtabellendatei. Vor dem Erstellen des Bohrbands erfolgt die Abfrage nach der Bohrungsklasse, über die die selektive Ausgabe durchkontaktierter, nicht durchkontaktierter, oder zu partiellen Durchkontaktierungen gehörender Bohrungen möglich ist.

Erstellen Sie mit den folgenden Kommandos die Bohrdaten für die auf dem Beispiellayout verwendete Bohrungsklasse **`Z`** mit einer Bohrertoleranz von 0.05mm (Bohrdatendatei **`demo.drl`**, Werkzeugtabellendatei **`demo.tol`**):



Durch die Eingabe eines Bindestrichs (`-`) auf die Abfrage nach der Bohrklasse werden diejenigen Bohrungen für die Ausgabe selektiert, die keiner speziellen Bohrklasse zugeordnet sind. Die erfolgreiche Ausgabe der Bohrdaten wird mit der Meldung **`Bohrdaten geschrieben.`** quittiert. Die Werkzeugtabelle **`demo.tol`** sollte nach Ausführung obiger Kommandos folgenden Inhalt haben:

```
/* Bohrer(Nummer) Bohrdurchmesser(mm) */
1     0.50
2     0.80
3     0.90
4     1.00
5     1.30
6     3.00
```

Die Bohrdatendatei **`demo.drl`** sollte wie folgt aussehen:

```
%
X1016Y762T6
X1016Y6350
M30
```

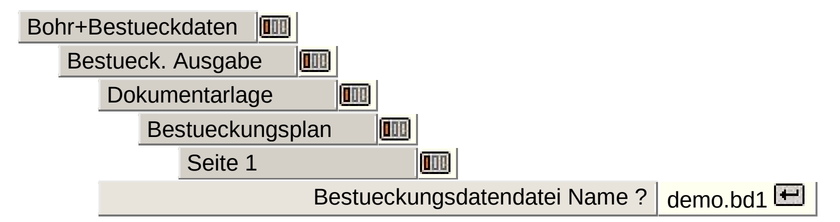
4.7.14 Bestückdaten

Die Bestückdaten werden mit Hilfe der auf einer Lage vorhandenen Texte generiert. Die Ausgabe besteht aus einer Liste der auf der für die Ausgabe selektierten Lage definierten Texte mit Koordinaten und Drehwinkel.

Um geeignete Ausgabedaten zu erhalten, ist es zweckmässig, eine eigene Dokumentarlage für die Bestückdaten zu verwenden. Auf dieser trägt man in den Bauteildefinitionen ein **$**-Zeichen für den Namen des Bauteils im Greifpunkt für den Bestückungsautomaten ein. Der Greifpunkt liegt im Ursprung des Koordinatenkreuzes, das beim Platzieren des Textes angezeigt wird.

Die Ausgabe der Bestückdaten erfolgt im Menü Bohr+Bestueckdaten mit der Funktion Bestueck. Ausgabe. Nach Aufruf dieser Funktion erscheint ein Untermenü, in dem die Ausgabelage gewählt werden kann. Beachten Sie die Option Mehrere Lagen. Damit können die Bestückdaten *mehrerer*, selektierbarer Lagen simultan ausgegeben werden. Die Option Mehrere Lagen aktiviert ein Popupmenü, in dem die auszugebenden Bestücklagen mit der linken oder rechten Maustaste selektiert bzw. deselektiert werden können. Der Col.-Button im Lagenauswahlmenü dient dazu, alle sichtbaren Lagen für die Ausgabe zu selektieren. Nach der Selektion der Ausgabelage(n) erfolgt die Abfrage nach dem Bestückdatenkanal.

Geben Sie mit den folgenden Kommandos die Bestückdaten für die Lage 1 des Beispiellayouts in die Datei **demo.bd1** aus:



Die erfolgreiche Ausgabe der Bestückdaten wird mit der Meldung

```
Bestueckungsdaten geschrieben.
```

angezeigt. Die Ausgabedatei sollte folgenden Inhalt haben:

```
/* Name X(mm) Y(mm) Winkel(Grad) */

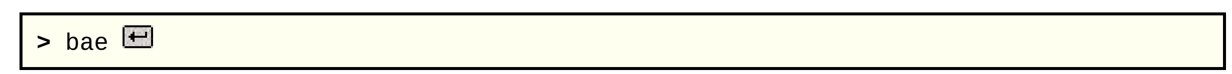
C100          4635    3651      0
C101          3619    3651      0
R104           984    5016    270
```

4.8 CAM-View

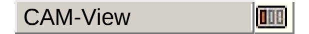
Der Programmteil **CAM-View** erfüllt drei Aufgabenstellungen. Dies ist zum ersten die visuelle Darstellung von Photoplotdaten im Gerberformat ("Gerber-Viewer"), von Bohrdaten im Sieb&Meier- bzw. Excellon-Format sowie von Fräsdaten im Excellon-Format. Des Weiteren können mehrere Gerber-, Bohr- und Fräsdatensätze mit verschiedenen Offsets bzw. Spiegelungsmodi eingelesen, selektiv verschoben und die gesamten so erhaltenen Daten in eine einzige Gerber-, Bohr- oder Fräsdatei geschrieben werden. Mit dieser Funktion können Nutzen, d.h. die Unterbringung mehrerer Layouts auf einem Film bzw. in einem Datensatz, generiert werden. Das dritte Aufgabengebiet von **CAM-View** ist das Erzeugen von Layouts aus den Gerberdaten von Fremdsystemen. Dabei werden aus den Gerberdaten Kupfer- und Dokumentarflächen generiert, die mit dem **Bartels AutoEngineer** weiterverarbeitet werden können.

4.8.1 Programmaufruf

Der Aufruf des **Bartels AutoEngineer** sollte grundsätzlich aus dem Verzeichnis erfolgen, in welchem die zu bearbeitenden Projektdateien abgelegt bzw. abzulegen sind. Wechseln Sie also zunächst in Ihr Projektverzeichnis. Zur Abarbeitung der in diesem Handbuch aufgeführten Beispiele ist es zweckmäßig, in das bei der Installation des **Bartels AutoEngineer** angelegte BAE-Jobs-Directory (d.h. in das Verzeichnis, in dem die Datei **`demo.ddb`** abgelegt ist) zu wechseln. Der Aufruf des **CAM-View**-Moduls erfolgt aus der Shell des **Bartels AutoEngineer**. Starten Sie diese von Betriebssystemebene aus mit folgendem Befehl:



Wählen Sie den Menüpunkt CAM-View mit der Maus an, und bestätigen Sie Ihre Wahl durch Drücken der linken Maustaste:



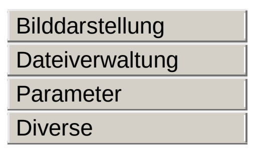
Nun wird das **CAM-View**-Modul des **AutoEngineer** geladen. Sollte der Programmaufruf fehlschlagen, dann deutet dies darauf hin, dass die Software nicht richtig installiert ist. Wir verweisen in diesem Zusammenhang auf die Bartels AutoEngineer® Installationsanleitung.

4.8.2 Hauptmenü

In der Benutzeroberfläche des **CAM-View**-Moduls werden neben bereits aus dem **Layouteditor** bekannten Menüs (wie Ansicht bzw. Bilddarstellung, Diverse) Funktionen zum Laden und Schreiben von Gerber- und Bohrdaten angeboten. Nach dem Aufruf von **CAM-View** befindet sich auf der rechten Bildschirmseite eine Menüleiste, bestehend aus dem Hauptmenü im oberen Bereich sowie dem im Hauptmenü selektierten Menü im unteren Bereich. Nach dem Laden von **CAM-View** ist das Menü Dateiverwaltung aktiviert, und der grüne Menübalken steht auf Gerberdaten laden.

Unter Windows und Motif kann anstelle der Standard- bzw. Seitenmenükonfiguration wahlweise auch ein Benutzerinterface mit Pulldownmenüs aktiviert werden. Hierzu ist mit Hilfe des Utilityprogramms **BSETUP** das Kommando **WINMENUMODE** mit der Option **PULLDOWN** in das Setup der BAE-Software einzuspielen (siehe hierzu auch Kapitel 7.2). Bei der Verwendung von Pulldownmenüs ist das Hauptmenü als horizontal ausgerichtete Menüleiste am oberen Ende der Benutzerschnittstelle angeordnet.

Das Hauptmenü ist während der Dauer der Arbeit mit dem **CAM-View** ständig verfügbar und ermöglicht die Aktivierung der folgenden Menüs:



Ansicht, Bilddarstellung

Im Menü Ansicht, das Sie außer durch Selektion im Hauptmenü auch immer über die mittlere Maustaste erreichen können, können Sie Zoomfunktionen aktivieren, das Eingabe- und Hintergrundraster definieren oder die Farbtabelle, die Farbzuordnung, den Zeichenmodus und den Koordinatenanzeigemodus einstellen.

Dateiverwaltung

Das Menü Dateiverwaltung enthält Funktionen zum Laden, Bewegen und Schreiben von Gerber-, Bohr- und Fräsdatensätzen, zum Erzeugen von Layouts aus geladenen Gerberdaten, sowie zum Löschen der aktuell im Speicher befindlichen CAM-Daten.

Parameter

Das Menü Parameter enthält Funktionen zum Setzen des Koordinatenversatzes und des Spiegelungsmodus für nachfolgende Einlesevorgänge, zur Festlegung des Via-D-Codes für die Erzeugung von Layouts aus Gerberdaten, zur Definition der Gerber-Blendentabelle, sowie zur Einstellung der Format- und Modusparameter für das Einlesen bzw. Schreiben von Gerberdaten.

Diverse

Im Menü Diverse kann der Programmabbruch oder der Rücksprung in die Shell des **Bartels AutoEngineer** veranlasst werden. Darüber hinaus enthält dieses Menü Funktionen zum Sortieren von Bohrungen, zur Anzeige einer Statistik über die in den aktuell geladenen Gerberdaten verwendeten Gerberblenden, sowie zur Anzeige von DDB-Dateiinhalten.

4.8.3 Modifizierte Benutzeroberfläche

Menübelegung und Tastaturprogrammierung

Einige der mit der BAE-Software installierten **User Language**-Programme definieren implizite **User Language**-Programmaufrufe über die eine weit reichend modifizierte Benutzeroberfläche mit einer Vielzahl von Zusatzfunktionen (Startups, Toolbars, Menübelegung, Tastaturprogrammierung) aktiviert wird. Das **User Language**-Startupprogramm **BAE_ST** wird automatisch beim Aufruf des **CAM-View**-Moduls gestartet. **BAE_ST** ruft seinerseits das **User Language**-Programm **UIFSETUP** auf, welches eine vordefinierte Menü- und Tastaturbelegung im **CAM-View**-Modul aktiviert. Änderungen bzw. Anpassungen der Menü- und Tastaturbelegung können *zentral* in der Quellcodedatei von **UIFSETUP** vorgenommen werden. Die aktuelle Tastaturbelegung kann mit dem **User Language**-Programm **HLPKEYS** angezeigt werden. Der Aufruf von **HLPKEYS** ist über die Funktion Tastaturbelegung aus dem Menü Hilfe möglich, sofern die vordefinierte Menübelegung aus **UIFSETUP** aktiviert ist. Mit dem **User Language**-Programm **UIFDUMP** kann die in der aktuellen Interpreterumgebung definierte Menü- und Tastaturbelegung in Form eines Reports angezeigt bzw. auf eine Datei ausgegeben werden. Mit dem **User Language**-Programm **UIFRESET** lässt sich die komplette Menü- und Tastaturbelegung zurücksetzen. **UIFSETUP**, **UIFDUMP** und **UIFRESET** sind auch über das Menü des **User Language**-Programms **KEYPROG** aufrufbar, welches zudem komfortable Funktionen zur Online-Tastaturprogrammierung sowie zur Verwaltung von Hilfstexten für **User Language**-Programme zur Verfügung stellt.

Kaskadierende Pulldownmenüs unter Windows/Motif

Die Windows- und Motifversionen des **CAM-View**-Moduls ermöglichen die Konfiguration kaskadierender Pulldownmenüs. Menüpunkte können ihrerseits implizit auf Untermenüs verweisen. Die Notwendigkeit des Aufrufs von Funktionen zur expliziten Anzeige von Untermenüs entfällt damit. Die Pulldownmenüs der Windows- und Motifversionen des **CAM-View**-Moduls werden über das **User Language**-Programm **UIFSETUP** entsprechend mit kaskadierenden Menüs ausgestattet. Untermenüfunktionen lassen sich damit einfach lokalisieren und starten. Die über die rechte Maustaste implementierte Wiederholfunktion ist entsprechend angepasst. Die Wiederholung von in Untermenus untergebrachten Funktionen vereinfacht sich dadurch erheblich.

Dialoge für Parametereinstellungen unter Windows/Motif

In den Windows- und Motifversionen des **CAM-View**-Moduls sind die folgenden Dialoge für Parametereinstellungen implementiert:

- Einstellungen - Einstellungen: Allgemeine **CAM-View**-Parameter
- Ansicht - Einstellungen: Bilddarstellungsparameter

In den Pulldownmenükonfigurationen werden die Standardfunktionen für Parametereinstellungen über das **User Language**-Programm **UIFSETUP** durch die obigen Menüfunktionen zum Aufruf der entsprechenden Dialoge ersetzt.

Pulldownmenükonfiguration unter Windows/Motif

Bei der Verwendung von Pulldownmenüs unter Windows und Motif wird über das **User Language**-Programm **UIFSETUP** eine an Windows angepasste Menüanordnung mit zum Teil geänderten Funktionsbezeichnungen und einer Vielzahl von Zusatzfunktionen konfiguriert. Das Hauptmenü des **CAM-View**-Moduls wird dabei wie folgt aufgebaut:



4.8.4 Grundsätzliches zur Bedienung

User Language

Im **CAM-View**-Modul ist der **Bartels User Language Interpreter** integriert, d.h. von **CAM-View** aus können **User Language**-Programme gestartet werden. Der Anwender hat damit die Möglichkeit, eigene Zusatzfunktionen nach anwender- bzw. firmenspezifischen Bedürfnissen zu implementieren und in das **CAM-View**-Modul einzubinden. Hierzu zählen zum Beispiel Statusanzeigen und Parametereinstellungen, Report- und Testfunktionen, firmenspezifische Batch-Prozeduren, usw. usf.

Im **CAM-View**-Modul können **User Language**-Programme explizit oder implizit aufgerufen werden. Der explizite Programmaufruf erfolgt über den Menüpunkt Anwenderfunktion im Menü Datei. Nach der Aktivierung dieses Menüpunktes ist auf die Abfrage nach dem Programmnamen der Name des aufzurufenden **User Language**-Programms (z.B. `ulprog`) explizit einzugeben. Die Betätigung einer beliebigen Maustaste oder die Eingabe eines Fragezeichens **?** auf die Abfrage nach dem Programmnamen bewirkt hierbei die Aktivierung eines Popupmenüs mit allen aktuell verfügbaren **User Language**-Programmen.

User Language-Programme können auch implizit über die Tastatur aktiviert werden. Diese Art des Programmaufrufs ist immer dann möglich, wenn nicht gerade eine andere interaktive Eingabe über Tastatur erwartet wird. Die Spezifikation des Programmnamens erfolgt dabei implizit durch Drücken einer Taste. Zulässige Tasten sind dabei die Standardtasten (1, 2, ..., 0, a, b, c, ...; entsprechende Programmnamen sind **cv_1**, **cv_2**, ..., **cv_0**, **cv_a**, **cv_b**, **cv_c**, ...) bzw. die Funktionstasten (F1, F2, ...; entsprechende Programmnamen sind dabei **cv_f1**, **cv_f2**, ...).

CAM-View ermöglicht darüber hinaus den ereignisgesteuerten Aufruf von **User Language**-Programmen. Dabei lösen spezielle Ereignisse bzw. Operationen implizit, d.h. automatisch den Aufruf von **User Language**-Programmen mit definierten Namen aus, sofern diese verfügbar sind. Im Einzelnen sind dies die **User Language**-Programme **CV_ST** beim Starten des **CAM-View**-Moduls, **CV_TOOL** bei Selektion eines Toolbarelements sowie **CV_ZOOM** bei Änderung des Zoomfaktors. Der Aufruf über die Startupsequenz der Interpreterumgebung eignet sich besonders zur automatischen Voreinstellung von modulspezifischen Parametern sowie zur Tastaturprogrammierung und Menübelegung. Bei Interaktionen in der Werkzeugliste werden die den selektierten Toolbarelementen zugewiesenen Funktionen ausgelöst. Die Änderung des Zoomfaktors kann dazu benutzt werden, Aktualisierungen in Funktionen zur Verwaltung von Entwurfsansichten auszulösen.

Mit der **Bartels User Language** werden darüber hinaus mächtige Systemfunktionen zur Tastaturprogrammierung und Menübelegung sowie zur Definition von Werkzeugleisten (Toolbars) zur Verfügung gestellt. Beachten Sie bitte, dass über die mit der BAE-Software ausgelieferten **User Language**-Programme eine Vielzahl von Zusatzfunktionen implementiert und transparent in die Benutzeroberfläche des **CAM-View**-Moduls eingebunden sind.

Eine ausführliche Beschreibung der **Bartels User Language** finden Sie im Bartels User Language Programmierhandbuch (Kapitel 4.2 enthält eine Auflistung aller mit der BAE-Software ausgelieferten **User Language**-Programme).

4.8.5 Bearbeiten von Gerberdaten

Allgemeine Parameter

Zur Festlegung von globalen Parametern für das Einlesen bzw. Schreiben von Datensätzen gibt es im Menü Einstellungen die Funktionen Einleseoffset, Gerber Format, Kreismodus, Spiegelung, Trailing/Leading Zeros, Extended Gerber, Optimierung und Koordinatenangabe.

Der Einleseoffset wird beim Einlesen von CAM-Daten zu den gelesenen Koordinaten hinzuaddiert. Durch Angabe von verschiedenen Einleseoffsets kann man z.B. das gleiche Layout an verschiedene Positionen laden und dann als ganzes wieder abspeichern (Nutzengenerierung). Die Bohrdaten und Gerberdaten verschiedener Lagen können gleichzeitig im Speicher gehalten werden. Dadurch muss der Einleseoffset für jedes Nutzenelement nur einmal angegeben werden. Vor der Generierung von Nutzen können mit Loeschen Speicher im Menü Datei eventuell im Speicher vorhandene Datensätze gelöscht werden.

Mit Gerber Format gibt man an, ob Gerber 2.3 Format (Voreinstellung; 1/1000 Inch), Gerber 2.4 Format (1/10000 Inch) oder ein Anderes Format bei der Erzeugung der Gerberdaten verwendet wurde. Bei der Wahl von Anderes Format ist die Länge einer Plottereinheit anzugeben. Damit lassen sich beliebige Gerberformate verarbeiten.

Mit Kreismodus kann die Verarbeitung von Kreissegmenten gesteuert werden. Bei Beliebige Kreisbögen (Defaulteinstellung) werden Kreisbögen ohne weitere Verarbeitung direkt eingelesen. Dies ist für die Ausgabedaten unseres **CAM-Prozessors** und die meisten gängigen Fremdsysteme sinnvoll. Einige Fremdsysteme können maximal Viertelkreisbögen ausgeben und bilden dabei den Betrag des Mittelpunktsvektors. Für solche Fälle muss die Einstellung Max. Viertelkreise gewählt werden.

Die Funktion Spiegelung im Menü Einstellungen ermöglicht die wahlweise Spiegelung der nachfolgend einzulesenden Datensätze. Mögliche Spiegelungsarten sind Keine Spiegelung (Voreinstellung), Spiegelung an X-Achse, Spiegelung an Y-Achse und Spiegelung an Ursprung. Beim Einlesen erfolgt die Spiegelung immer *vor* der Verschiebung um den Einleseoffset.

Die Funktion Trailing/Leading Zeros aus dem Menü Einstellungen dient der wahlweisen Berücksichtigung der Unterdrückung führender bzw. endender Nullen beim Einlesen von Gerberkoordinaten. Voreingestellt ist der Bearbeitungsmodus Trailing Zeros zur Beibehaltung endender Nullen und Unterdrückung führender Nullen. Der Modus Leading Zeros zur Beibehaltung führender und Unterdrückung endender Nullen ist nur für die beiden Standardgerberformate 2.3 und 2.4 wirksam, d.h. bei Angabe anderer Konvertierungsfaktoren wird immer im Modus Trailing Zeros gearbeitet.

Über die Funktion Extended Gerber kann wahlweise eine Fotoplotausgabe im Format RS-274-X (Extended Gerber with Embedded Apertures) erfolgen. Beim Einlesen von Gerberdaten wird das Format RS-274-X automatisch erkannt.

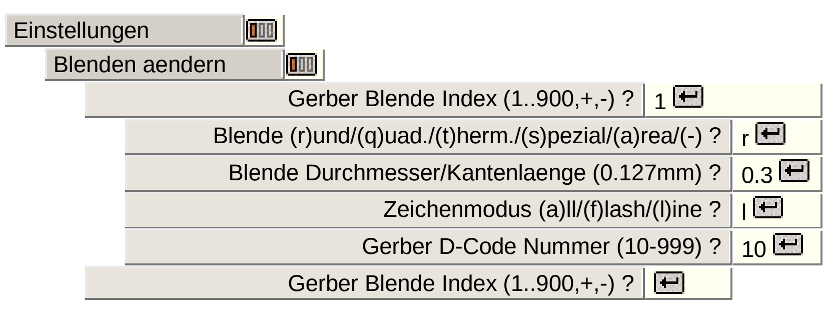
Mit der Funktion Optimierung kann wahlweise eine optimierte Gerberausgabe erzeugt werden. Dabei werden gleichbleibende Koordinatenkomponenten und die Wiederholung des Kommandos D01 ("Licht aus") bei längeren Linienzügen unterdrückt, wodurch sich eine signifikante Reduzierung der Plotdatenmengen ergibt. Beim Einlesen werden optimierte Gerberdaten automatisch erkannt.

Über die Funktion Koordinatenangabe kann angegeben werden, ob die aus Fremdsystemen einzulesenden Gerber- bzw. Excellonkoordinaten als Absolutkoordinaten (Defaultoption Abs.-Koordinaten) oder als relative bzw. inkrementale Koordinaten (Option Inkr.-Koordinaten) zu interpretieren sind.

Blendentabellen

Für die Bilddarstellung wird die beim Plot verwendete Blendentabelle benötigt. Diese kann im **CAM-Prozessor** erstellt und abgespeichert werden. **CAM-View** lädt beim Start die Blendentabelle mit dem Namen **`standard`**. Andere Blendentabellen können mit Blenden laden im Menü Datei geladen werden. Das Laden von Blendentabellen kann auch erfolgen, *nachdem* Gerberdaten eingelesen wurden; in diesem Fall ändert sich die Bilddarstellung entsprechend den neuen Blendentabellenvorgaben. Einzelne Einträge der aktuell verwendeten Blendentabelle können mit der Funktion Blenden aendern im Menü Einstellungen jederzeit geändert werden. Eine so geänderte Blendentabelle kann mit Blenden speichern im Menü Datei abgespeichert werden.

Tragen Sie unter D-Code 10 mit den folgenden Kommandos eine runde Line-Blende mit 0.3mm Durchmesser in den ersten Blendentabellenplatz ein:

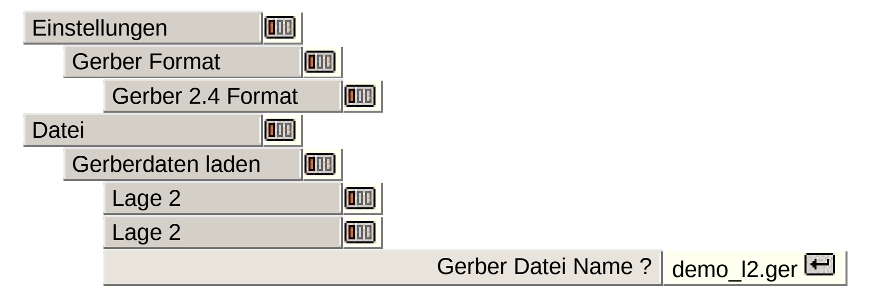


Einlesen von Gerberdaten

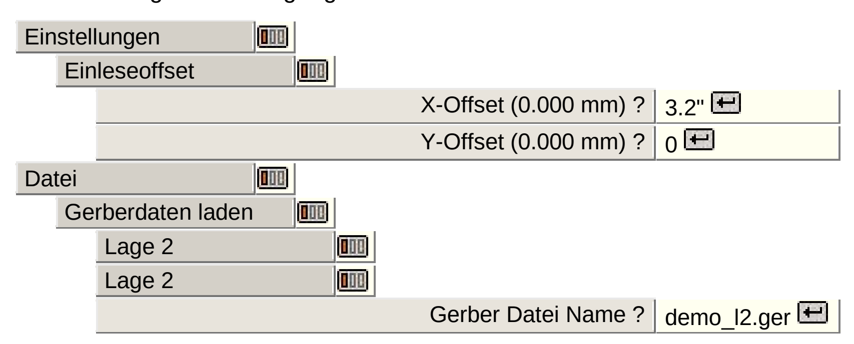
Das Einlesen der Daten erfolgt mit Gerberdaten Laden im Menü Datei. Vor der Abfrage für den Dateinamen wird noch die Angabe einer Lage für liniengezogene Strukturen und anschließend die Angabe einer Lage für geblitzte Strukturen verlangt. Dies wird für das Einlesen von in Fremdsystemen erzeugten Gerberdateien benötigt, damit aus den verschiedenen Lagen zugeordneten Daten bzw. Strukturen später ein Layout erzeugt werden kann. Die Lageninformation wird auch bei der Generierung von Nutzen benötigt um die Daten der einzelnen Lagen in getrennte Datenfiles schreiben zu können.

Man kann die Lagenzuordnung auch dazu verwenden, um verschiedene Gerberdateien (desselben Layouts) miteinander zu vergleichen. Lädt man die erste Gerberdatei auf Lage 1 und die zweite Gerberdatei auf Lage 2, so erscheinen durch die Mischfarbendarstellung (bei Farbzuordnung nach Lagen) die beiden Versionen gemeinsamen Strukturen in der aus den Farben für Lage 1 und 2 resultierenden Mischfarbe.

Stellen Sie mit den folgenden Kommandos Gerber 2.4 Format ein, und laden Sie die in Kapitel 4.7.11 erzeugte Gerberdatei **`demo_l2.ger`** in den Arbeitsspeicher (liniengezogene Strukturen auf Lage 2, geblitzte Strukturen auf Lage 2):



Wiederholen Sie obigen Ladevorgang mit einem Einlese-Offset von 3.2 Zoll in X-Richtung:



Ansicht, Bilddarstellung

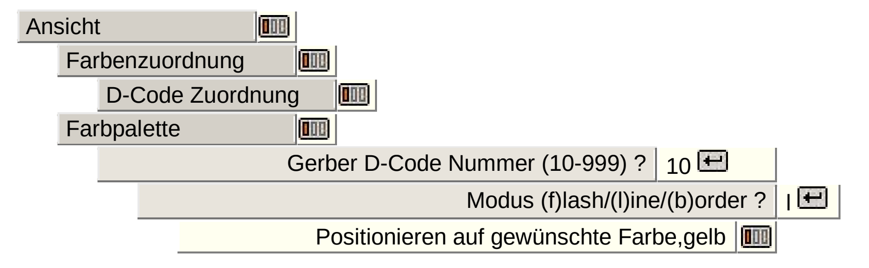
Zur Bilddarstellung stehen im Menü Ansicht die schon vom **Layouteditor** her bekannten Zoomfunktionen zur Auswahl des Bildausschnittes und des Hintergrundrasters zur Verfügung.

Für die Gerberdatendarstellung kann zwischen den beiden Modi Flaechendarstellung und Randdarstellung gewählt werden. Bei Randdarstellung wird zusätzlich zur Flächendarstellung um jedes Element eine Randlinie gezogen. Dies ist z.B. für die Betrachtung liniengefüllter Flächen nützlich.

Es stehen zwei Methoden zur Farbauswahl zur Verfügung. Bei der ersten wird die Farbe eines Elements durch den D-Code und den Zeichenmodus bestimmt. Bei der zweiten Methode werden die Elemente in der Farbe der Lage, auf die sie geladen wurden, dargestellt. Zwischen diesen beiden Methoden kann im Menüpunkt Farbenzuordnung mit D-Code Zuordnung (Defaulteinstellung) und Lagenzuordnung ausgewählt werden.

Abhängig von der Farbzuordnung erscheint bei Aufruf des Menüpunktes Farbpalette entweder die Aufforderung nach Eingabe eines D-Codes und eines Zeichenmodus oder ein Lagenauswahlmenü. Die verschiedenen Zeichenmodi sind `f` (Flash) für geblitzte Strukturen, `l` (Line) für liniengezogene Strukturen und `b` (Border) für die Umrandung von Zeichenflächen. Die Defaulteinstellung der Farben für die D-Codes ist hellgrau für geblitzte Strukturen, mittelgrau für liniengezogene Strukturen und rot für die Umrandungslinien.

Stellen Sie mit den folgenden Kommandos die Farbe für die Elemente, die mit D-Code 10 (Line) generiert wurden, auf gelb ein:



Wird die Farbe schwarz gewählt, so werden die entsprechenden Elemente nicht gezeichnet.

Report

Mit der Funktion Report im Menü Diverse kann eine Statistik über die verwendeten Blenden abgerufen werden. Angezeigt wird eine Liste der verwendeten D-Codes mit der jeweiligen Angabe, wie viele Strukturen in welchem Modus generiert wurden.

Rufen Sie die Funktion Report auf:



Das System sollte nun im Grafikarbeitsbereich folgende Liste ausgeben:

```
D10  0.30 mm  rund :
      Linien  : 58
D11  0.25 mm  rund :
      Linien  : 454
D16  1.52 mm  rund :
      Geblitzt: 64
D18  2.54 mm  rund :
      Geblitzt: 16
      Linien  : 4
D29  1.40 mm  quad :
      Geblitzt: 20 ...
```

Vom Report ausgegebene Strukturen mit der Modus-Angabe `Linien mit Länge null` sind Positionen, die zwar angefahren werden, aber bei denen keine Belichtung erfolgt.

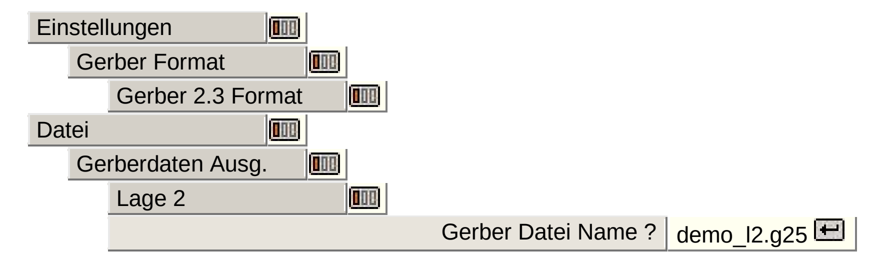
Bewegen von Gerberdaten

Mit der Funktion Datensatz bewegen aus dem Menü Datei können zuvor eingelesene Gerberdatensätze nachträglich verschoben werden. Der gewünschte Datensatz wird in einem Popupmenü ausgewählt. Die Datensätze werden darin in der Einlesereihenfolge dargestellt. In jeweils einer Zeile pro Datensatz werden die Lagen, der Offset gegenüber den Ursprungsdaten, der Spiegelungsmodus und der Dateiname der Eingabedaten angezeigt. Nach Wahl des Datensatzes kann durch Selektion von Anfangs- und Endpunkt der Verschiebung ein Verschiebungsvektor angegeben werden. Das Eingaberaster bei der Verschiebung ist gleich dem Hintergrundraster der Bilddarstellung. Während der Verschiebung kann über die rechte Maustaste ein Menü zur Eingabe von absoluten und relativen Sprüngen aktiviert werden.

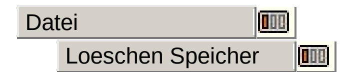
Schreiben von Gerberdaten

Die im Speicher befindlichen Gerberdaten einer Lage können mit Gerberdaten Ausg. unter Angabe eines Dateinamens abgespeichert werden.

Geben Sie mit folgenden Kommandos die aktuell auf Signallage 2 geladenen Gerberdaten im Gerber 2.3 Format auf die Datei **`demo_l2.g25`** aus:



Nun können die im Speicher befindlichen Daten mit Loeschen Speicher wieder entladen werden, um neue Daten zu lesen:



4.8.6 Bearbeiten von Bohr- und Fräsdaten

Einlesen von Bohr- bzw. Fräsdaten

Das Einlesen von Bohr- bzw. Fräsdaten erfolgt mit der Funktion Bohrdaten laden aus dem Menü Datei. Vor dem Laden von Bohrdaten im Format Sieb&Meier muss auf jeden Fall erst eine Werkzeugtabelle geladen werden. Dies geschieht mit der Funktion Werkzeugtab. laden aus dem Menü Datei. Werden mehrere Datensätze hintereinander geladen, so ist zu beachten, dass die Werkzeugtabellen übereinstimmen, da sonst beim Schreiben keine brauchbaren Ausgabedaten erzeugt werden können. Weiterhin ist zu beachten, dass die Funktion Werkzeugtab. laden alle zuvor geladenen Bohr- bzw. Fräsdaten aus dem Hauptspeicher löscht.

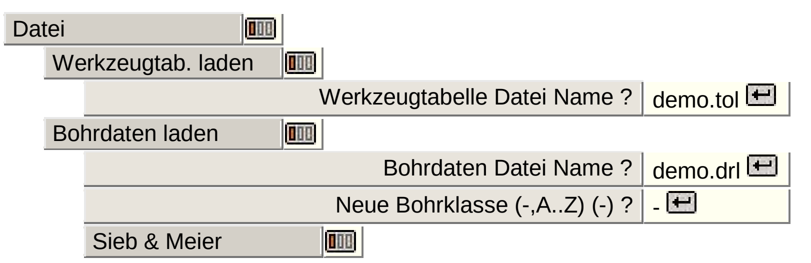
Beim Einlesen und Abspeichern von Bohrdaten kann zwischen den Formaten Sieb&Meier und Excellon gewählt werden. Die Koordinateneinheiten sowie die Behandlung führender und endender Nullen werden aus der Eingabedatei abgeleitet. Sind in der Eingabedatei keine Formatbefehle vorhanden, so werden die Einstellungen der Gerbereingabe übernommen. Bei Spezifikation von Inch-Einheiten in Excellon-Eingabedatensätzen werden die Koordinatenangaben automatisch entsprechend der aktuellen Gerber-Formatspezifikation (Gerber Format 2.3, Gerber Format 2.4, etc.) interpretiert.

Beim Einlesen von Daten im Excellon-Format werden Fräsdaten automatisch als solche erkannt und verarbeitet. Beim Schreiben von Bohrdaten im Excellon-Format muss angegeben werden, ob Fräs- oder Bohrdatenausgabe erwünscht ist (es könnten innerhalb einer Bohrklasse sowohl Fräs- als auch Bohrdaten vorhanden sein).

Die Bohr- und Fräsdatensätze werden mit Angabe der Bohrklasse geladen und gespeichert. Dadurch ist es möglich, verschiedene Bohr- bzw. Fräsdatensätze mit unterschiedlichen Bohrklassen zu laden und wieder nach Bohrklassen getrennt zu speichern. Bohrungen bzw. Fräsereinstechpunkte, die nicht der Bohrklasse **`-`** angehören, werden bei der Bilddarstellung mit dem entsprechenden Buchstaben für die Bohrklasse (**`A`**, **`B`**, ..., **`Z`**) angezeigt.

Beim Laden von Bohrdaten in das **CAM-View**-Modul wird automatisch der Farbtabelleneintrag für Bohrungen aktiviert. Ist keine Farbe für Bohrungen definiert, so wird automatisch ein weisser Farbtabelleneintrag gesetzt. Damit wird sichergestellt, dass neu geladene Bohrdatensätze in jedem Fall visualisiert werden, und der Erfolg des Ladevorgangs somit unmittelbar kontrolliert werden kann.

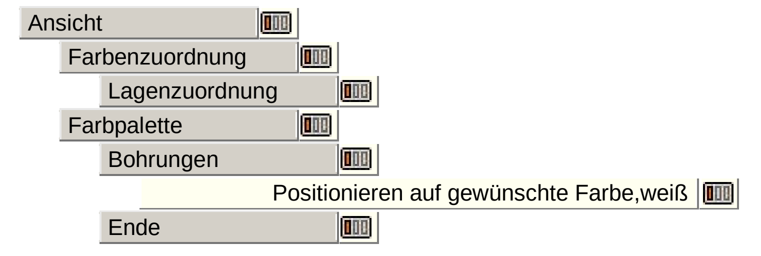
Laden Sie mit den folgenden Kommandos die in Kapitel 4.7.12 im Format Sieb&Meier erzeugte Bohrdatendatei **`demo.drl`** mit zugehöriger Werkzeugtabelle **`demo.tol`** unter der Bohrklasse **`-`** in den Arbeitsspeicher:



Ansicht, Bilddarstellung

Zur Bilddarstellung stehen im Menü Ansicht die schon vom **Layouteditor** her bekannten Zoomfunktionen zur Auswahl des Bildausschnittes und des Hintergrundrasters zur Verfügung.

Die Anzeige der aktuell geladenen Bohr- bzw. Fräsdaten erfolgt über die Selektion einer Farbe mit dem Eintrag Bohrungen aus dem Menü Farbpalette bei Einstellung der Farbzuordnung auf Lagenzuordnung. Selektieren Sie mit den folgenden Kommandos die Farbe weiß für die Darstellung der Bohr- und Fräsdaten:



Bewegen von Bohr- bzw. Fräsdaten

Mit der Funktion Datensatz bewegen aus dem Menü Datei können zuvor eingelesene Bohr- bzw. Fräsdatensätze nachträglich verschoben werden. Der gewünschte Datensatz wird in einem Popupmenü ausgewählt. Die Datensätze werden darin in der Einlesereihenfolge dargestellt. In jeweils einer Zeile pro Datensatz werden die Lagen, der Offset gegenüber den Ursprungsdaten, der Spiegelungsmodus und der Dateiname der Eingabedaten angezeigt. Nach Wahl des Datensatzes kann durch Selektion von Anfangs- und Endpunkt der Verschiebung ein Verschiebungsvektor angegeben werden. Das Eingaberaster bei der Verschiebung ist gleich dem Hintergrundraster der Bilddarstellung. Während der Verschiebung kann über die rechte Maustaste ein Menü zur Eingabe von absoluten und relativen Sprüngen aktiviert werden.

Schreiben von Bohr- bzw. Fräsdaten

Die im Speicher befindlichen Bohr- bzw. Fräsdaten können mit Bohrdaten Ausgabe im Menü Datei in eine Datei geschrieben werden.

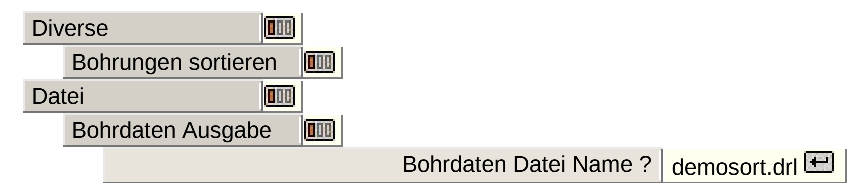
Geben Sie die aktuell geladenen Bohrdaten mit den folgenden Kommandos im Excellon-Format auf die Datei **`demo.exc`** aus:



Sortieren von Bohrdaten

Im Menü Diverse steht unter dem Menüpunkt Bohrungen sortieren ein einfacher Algorithmus zur Sortierung der im Speicher befindlichen Bohrungen zur Verfügung. Dabei wird innerhalb jedes einzelnen der aktuell geladenen Bohrdatensätze *gesondert* sortiert. Wird die Sortierung über mehrere Bohrdatensätze gewünscht, so sind diese zunächst einzulesen und dann als ein Datensatz abzuspeichern. Nach Loeschen Speicher und Einlesen des neuen Datensatzes können die Bohrungen in ihrer Gesamtheit sortiert werden. Weiterhin ist anzumerken, dass der Rechenzeitaufwand quadratisch mit der Anzahl der Bohrungen pro Bohrwerkzeug wächst, was bei sehr großen Layouts (Anzahl Bohrungen >1000) zu erheblichen Rechenzeiten führen kann.

Sortieren Sie nun die im Speicher befindlichen Bohrdaten und schreiben Sie sie in die Datei **`demosort.drl`**:



4.8.7 Erzeugen von Layouts aus Gerberdaten

Laden von Gerberdaten

Zum Erzeugen von Layouts müssen zunächst die Gerberdaten der einzelnen Lagen eingelesen werden. Dabei gibt man zweckmäßigerweise für die liniengezogenen Strukturen die entsprechende Signallage an. Die geblitzten Strukturen stellen meist Pads dar und sollten daher als Platzierungshilfe z.B. auf die Dokumentarlage für den Bestückungsplan gelegt werden. Eine noch bessere Platzierungshilfe stellen natürlich im Gerberformat erstellte Bestückungspläne dar, die ebenfalls auf die entsprechende Dokumentarlage geladen werden können.

Layout erzeugen

Sind alle Gerberdaten in den Speicher geladen, so kann mit Layout erzeugen die Ausgabe des Layouts gestartet werden. Vor der Angabe des Datei- und Elementnamens, in dem das Layout abzulegen ist, muss der Benutzer noch einen D-Code für Vias spezifizieren. Mit diesem D-Code geblitzte Strukturen werden bei der Ausgabe in Vias umgesetzt. Da Vias auf allen Lagen vorhanden sind, werden diese nur für das erste eingelesene Datenfile generiert. Mit dem Via-D-Code geblitzte Strukturen aus den nachfolgend eingelesenen Datenfiles werden bei der Ausgabe ignoriert.

Für das Via wird in der Layoutdatei automatisch ein entsprechender Padstack mit einem Pad der Blendengröße auf allen Lagen und einer Bohrung mit halber Blendengröße generiert.

Weitere Bearbeitung des Layouts

Bei der Bearbeitung des erzeugten Layouts ist es wichtig, die nachfolgend aufgeführte Bearbeitungsreihenfolge unbedingt einzuhalten, um brauchbare Daten zu erhalten.

Das erzeugte Layout wird zunächst in den **Layouteditor** geladen. Danach platziert man mit Neues Bauteil (Menü Bauteile) nacheinander die Bauteile auf dem Layout. Nach Platzierung aller Bauteile ist das Layout abzuspeichern. Dann wird mit der Funktion Rueck-Netzliste (Menü Diverse) eine Netzliste mit dem gleichen Namen wie der Layoutplan erzeugt. Rueck-Netzliste generiert aus dem auf der Leiterkarte befindlichen Kupfer eine Netzliste, die direkt in der angegebenen DDB-Datei (nicht jedoch im Arbeitsspeicher) abgelegt wird. Danach sollte das Layout gleich wieder neu geladen werden. Auf keinen Fall dürfen Aktionen durchgeführt werden, die zu einem Abspeichern des aktuell geladenen Layouts führen können, da dadurch wieder eine leere Netzliste in die Datei geschrieben wird. Unmittelbar nach dem Laden des Layouts führt das System automatische eine Connectivity-Generierung durch, und anschließend kann das Layout weiter bearbeitet werden.

Kapitel 5
IC-/ASIC-Entwurf

HINWEIS
Die in diesem Kapitel beschriebenen Softwaremodule sind nur in Bartels AutoEngineer IC Design verfügbar.

Dieses Kapitel erläutert die Handhabung der Programm-Module **Chipeditor (IC-Maskeneditor)**, **Cellplacer** und **Cellrouter** für den physikalischen Entwurf integrierter Schaltungen (ICs) und ASICs. Darüberhinaus werden die Module **GDS-View** und **CIF-View** zum Importieren und Prüfen von Zellenbibliotheken und IC-Maskendaten in den Formaten GDS und CIF vorgestellt. In diesem Kapitel können wir leider keine realen Designbeispiele zeigen, da sowohl die Parameter für den IC-Fertigungsprozess als auch die Zellenbibliotheken vom IC-Hersteller bereitgestellt werden. Ihr IC-Hersteller wird Ihnen diese Daten sicherlich kostenfrei in einem zur Übernahme in den **Bartels AutoEngineer** geeigneten Format (z.B. GDS) zur Verfügung stellen, aber die Veröffentlichung derartiger Daten verbietet sich in allen uns bekannten Fällen aufgrund von NDAs ("Non Disclosure Agreements").

Inhalt

Kapitel 6
Neuronales Regelsystem

Dieses Kapitel beschreibt das im **Bartels AutoEngineer** integrierte neuronale Regelsystem, d.h. die Bartels Rule Specification Language zur Definition neuronaler Regeln, die Handhabung des **Bartels Rule System Compilers** zur Kompilierung von Regeldefinitionen, sowie die Möglichkeiten der Anwendung von neuronalen Regeln im BAE-Designprozess.

Inhalt

6.1 Allgemeine Hinweise

Im **Bartels AutoEngineer** ist ein neuronales Regelsystem integriert. Damit ist es möglich, einzelne oder mehrere Regeln bzw. Regelsätze zu definieren und diese individuellen Objekten im **AutoEngineer** zuzuweisen. Damit lassen sich Attribute zur Steuerung von Entwurfsabläufen wie z.B. Vorgaben für die Platzierung bestimmter Bauteiltypen (Einschränkungen hinsichtlich Drehung oder Spiegelung), lagenspezifische Mindestabstände für das Autorouting, netz- bzw. netzgruppenspezifische Vorgaben für das Verlegen von Leiterbahnen (maximale bzw. minimale Leiterbahnlänge, maximale Parallelführung, usw.) festlegen. Darüber hinaus lassen sich mit Hilfe von Regeldefinitionen auch komplexere Verfahrensabläufe zur Lösung spezieller Probleme festlegen, die über das Regelsystem automatisiert angewendet werden können. Beispiele hierfür sind die Durchführung spezieller Entwurfsregelprüfungen (EMV, Analogdesign, Hochfrequenztechnik, etc.). oder **Autorouter**-Läufe mit entsprechend der Problemstellung ausgewählten Strategieparametersätzen bzw. Optionsvorgaben.

Die Spezifikation von Regeln erfolgt über eine Prolog-ähnliche Programmiersprache mit speziellen Operatoren zur komfortablen Ermittlung optimaler (bzw. nahezu optimaler) Lösungen zu spezifischen Regelabfragen bzw. Ausgabeanforderungen. Zur Übersetzung von Regeldefinitionen steht ein spezieller Rule System Compiler zur Verfügung. Kompilierte Regeln werden entweder automatisch durch spezielle interne BAE-Systemfunktionen angewendet oder können mit Hilfe anwenderprogrammierter **User Language**-Programme aktiviert werden.

Regeln, die nur einen einzigen Prädikatwert setzen, können auch dynamisch ohne Umweg über eine `.rul`-Datei generiert und an Elemente zugewiesen werden. Dies vereinfacht den Umgang mit dem Regelsystem erheblich. Die im System unterstützten Regeln sind für den Benutzer transparent über menüintegrierte **User Language**-Programme verarbeitbar. Eine eingehende Kenntnis der Arbeitsweise des **Neuronalen Regelsystems** ist für diese Art der Regelsystemanwendung nicht erforderlich.

6.2 Regeldefinition

6.2.1 Bartels Rule Specification Language

Die Bartels Rule Specification Language zur Definition von Regeln bzw. Regelsätzen ist ähnlich der Programmiersprache Prolog, beinhaltet jedoch spezielle Operatoren, über die nicht nur alle möglichen, sondern die optimalen Lösungen zu einer spezifischen Regelabfrage bzw. Ausgabeanforderungen gefunden werden können. Regeln können für individuelle Objekte wie z.B. Bauteile, Netze, Leiterbahnen, usw. definiert werden. Darüber hinaus unterstützt das Regelsystem auch die Anwendung komplexer Verfahren wie z.B. die Durchführung spezieller Designregelprüfungen oder die Aktivierung von **Autorouter**-Läufen mit automatisch an die Problemstellung angepassten Strategieparametersätzen. Das Regelsystem arbeitet bei der Regelauswertung mit neuronalen Netzen, um auch in mehrdimensionalen Lösungsräumen eine hinreichend schnelle Fokussierung auf einen möglichst optimalen Lösungweg zu gewährleisten.

6.2.2 Bartels Rule System Compiler

Das Utilityprogramm **RULECOMP** ist der Compiler zur Übersetzung von Quelltexten zur Spezifikation von Regeln für das im **Bartels AutoEngineer** integrierte **Neuronale Regelsystem**. Nähere Informationen zu **RULECOMP** finden Sie auch in Bartels AutoEngineer Benutzerhandbuch - Kapitel 7.14.

Kompilierte Regeln werden entweder automatisch durch spezielle interne BAE-Systemfunktionen angewendet oder können mit Hilfe anwenderprogrammierter **User Language**-Programme aktiviert werden.

6.3 Regelsystemanwendungen

Im **Bartels AutoEngineer** sind eine Vielzahl mächtiger Zusatzfunktionen über das integrierte **Neuronale Regelsystem** implementiert. Dieses Kapitel enthält eine Übersicht über die im Schaltplanpaket und im Layoutsystem bereitgestellten Regelsystemanwendungen.

6.3.1 Regelsystemanwendungen für den Schaltungsentwurf

Antennenhighlight

Über das Regelsystem kann gesteuert werden, ob die Pins von Netzen mit im Leeren endenden Einzelsegmentverbindungen (sogenannten Antennen), gehighlightet werden sollen. Üblicherweise werden derartige Pins nicht gehighlightet, wenn die angeschlossene Einzelsegmentverbindung nur kurz ist ("Funktion" zum Abhaken nicht anzuschließender Pins). Zur Aktivierung des Antennenhighlightmodus ist die Regel **`scm_pin_drc`** (siehe Regeldefinitionsdatei **`scm.rul`** im **User Language**-Verzeichnis **`baeulc`**) an das aktuell geladene SCM-Element bzw. das aktuell bearbeitete Projekt zuzuweisen. Die Zuweisung von Regeln an SCM-Elemente kann mit dem **User Language**-Programm **SCMRULE** vorgenommen werden. **SCMRULE** kann entweder über Anwenderfunktion oder (sofern das **User Language**-Programm **UIFSETUP** aktiviert wurde) über die Funktion Regelzuweisungen aus dem Menü Einstellungen aufgerufen werden.

Warnung

Das Setzen des Antennenhighlightmodus ist nur möglich, wenn die Regeldefinitionsdatei **`scm.rul`** aus dem **User Language**-Verzeichnis und damit die Regel **`scm_pin_drc`** mit dem Rule System Compiler **RULECOMP** kompiliert wurde. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei **`brules.vdb`** im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner muss die Regel **`scm_pin_drc`** auch auf diesem bereitgestellt bzw. kompiliert werden, damit der Antennenhighlightmodus gesetzt bzw. berücksichtigt werden kann.

Busdarstellungsmodus

Über das Regelsystem ist es möglich, Busse auf Schaltplanebene wahlweise ausgefüllt darzustellen bzw. zu plotten. Hierzu wird geprüft, ob die Regel **`scm_bus_fill`** (siehe Regeldefinitionsdatei **`scm.rul`** im **User Language**-Verzeichnis **`baeulc`**) aktiviert, d.h. an das aktuell geladene SCM-Element zugewiesen wurde. Die Zuweisung von Regeln an SCM-Elemente kann mit dem **User Language**-Programm **SCMRULE** vorgenommen werden. Zum bequemen Setzen des Busdarstellungsmodus kann wahlweise auch das **User Language**-Programme **SCMCON** verwendet werden. **SCMCON** aktiviert ein Auswahlmenü mit den Optionen Randdarstellung und Fuelldarstellung. **SCMCON** kann entweder über Anwenderfunktion oder (sofern das **User Language**-Programm **UIFSETUP** aktiviert wurde) über die Funktion Busdarstellung aus dem Menü Verbindungen aufgerufen werden.

Warnung

Das Setzen des Busdarstellungsmodus ist nur möglich, wenn die Regeldefinitionsdatei **`scm.rul`** aus dem **User Language**-Verzeichnis und damit die Regel **`scm_bus_fill`** mit dem Rule System Compiler **RULECOMP** kompiliert wurde. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei **`brules.vdb`** im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner muss die Regel **`scm_bus_fill`** auch auf diesem bereitgestellt bzw. kompiliert werden, damit der Busdarstellungsmodus gesetzt bzw. berücksichtigt werden kann.

Textklassen zur Steuerung der Textsichtbarkeit

Das Untermenü Weitere Funktionen aus dem Menü Texte enthält die Funktion Textklasse mit den Optionen Text zuweisen, Symbolmaske und Planmaske. Damit können Textklassen zur Steuerung der Textsichtbarkeit für die Bilddarstellung und Plotausgabe an Texte zugewiesen werden. Die Option Text zuweisen aktiviert eine Dialogbox zur Zuweisung der Klassenzugehörigkeit mausselektierbarer Texte. Mit der Option Symbolmaske können auf Schaltplanebene für mausselektierbare Symbole Texte anhand der Klassenzugehörigkeit ausgeblendet werden. Mit der Option Planmaske können alle Texte eines Schaltplanes anhand der Klassenzugehörigkeit ausgeblendet werden. Ein Text kann mehreren Klassen zugewiesen werden. Die Ausblendung des Textes erfolgt, sobald eine der dem Text zugewiesenen Klassen ausgeblendet wird. Texte, die keiner Klasse angehören, sind immer sichtbar.

Es werden bis zu 31 verschiedene Textklassendefinitionen unterstützt. Für diese Textklassen können in der Datei **bae.ini** wahlweise Textklassennamen vorgegeben werden. Ohne Textklassendefinitionen werden automatisch die Klassennamen **Class 1**, **Class 2** usw. zugewiesen.

Die Textklassen sind insbesondere bei Verwendung von Symbolen mit einer Vielzahl von Attributen nützlich. Durch Ausblendung einzelner Attribute, die für das Lesen des Schaltplanes von geringerer Bedeutung sind, laesst sich die Planlesbarkeit steigern.

Textsichbarkeit auf rotierten Symbolen

Über die Regeln **scm_rot_vis_0**, **scm_rot_vis_90**, **scm_rot_vis_180** und **scm_rot_vis_270** aus der Regeldefinitionsdatei **scm.rul** im User Language-Directory (**baeulc**) kann die Sichtbarkeit von Texten in Abhängigkeit von der Symbolrotation beim Platzieren gesteuert werden. Texte, die die Regel **scm_rot_vis_0** zugewiesen bekommen haben, werden nur bei einer Platzierung des Symbols mit 0 Grad angezeigt, usw. Mit diesen Regel können Attributwerte oder Namensreferenzen für verschiedene Drehwinkel des Symbols an verschiedenen Stellen platziert werden. Zur einfachen Editierbarkeit dieser Regeln stehen im **User Language**-Programm **SCMRULE** auf Symbolebene die beiden Optionen Rotation aufloesen und Rotation zusammenfassen zur Verfügung. Rotation aufloesen erzeugt aus dem aktuell geladenen Symbol vier Kopien mit Drehwinkel 0, 90, 180 und 270 Grad. In diesen Kopien können die Texte bearbeitet werden. Mit Rotation zusammenfassen werden die Texte wieder auf ein einzelnes Symbol kopiert und automatisch mit den entsprechenden Regeln versehen. **SCMRULE** kann entweder über Anwenderfunktion oder (sofern das **User Language**-Programm **UIFSETUP** aktiviert wurde) über die Funktion Regelzuweisungen aus dem Menü Einstellungen aufgerufen werden.

Warnung

Die Zuweisung der Regeln zur Textsichtbarkeit auf rotierten Symbolen ist nur möglich, wenn die Regeldefinitionsdatei **scm.rul** aus dem **User Language**-Verzeichnis und damit die Regeln **scm_rot_vis_0**, **scm_rot_vis_90**, **scm_rot_vis_180** und **scm_rot_vis_270** mit dem Rule System Compiler **RULECOMP** kompiliert wurden. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei **brules.vdb** im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner müssen die Regeln auch auf diesem bereitgestellt bzw. kompiliert werden, damit diese Regeln aktiviert bzw. berücksichtigt werden können.

Plotten von Pinmarkergrafiken

Grafiken (Grafiklinien, Punktlinien, Grafikflächen) auf Markerebene werden bei der Plotausgabe üblicherweise unterdrückt. Durch Zuweisung der Regel **scm_pin_marker_plot** aus der Regeldefinitionsdatei **scm.rul** im User Language-Directory (**baeulc**) an das aktuell geladene SCM-Element kann die Plotausgabe von Pinmarkergrafiken erzwungen werden. Die Zuweisung von Regeln an SCM-Elemente kann mit dem **User Language**-Programm **SCMRULE** vorgenommen werden.

Warnung

Die Zuweisung der Regel **scm_pin_marker_plot** ist nur möglich, wenn die Regeldefinitionsdatei **scm.rul** aus dem **User Language**-Verzeichnis und damit die Regel **scm_pin_marker_plot** mit dem Rule System Compiler **RULECOMP** kompiliert wurde. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei **brules.vdb** im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner muss die Regel **scm_pin_marker_plot** auch auf diesem bereitgestellt bzw. kompiliert werden, damit die Plotausgabe von Pinmarkergrafiken aktiviert werden kann.

Steuerung der Plotsichtbarkeit

Die Funktionen Gruppe setzen und Elem. selektieren aus dem Untermenü Regelzuweisungen des Menüs Einstellungen enthalten Optionen zum Setzen bzw. Selektieren nach der Plotsichtbarkeit. Per Voreinstellung sind mit Ausnahme von Kommentartexten und Tagsymbolen alle Strukturen für den Plot sichtbar geschaltet. Durch Zuweisung des Attributes Nicht Plotten können einzelne Elemente für die Plotausgabe ausgeblendet werden. Die entsprechenden Elementstrukturen werden am Bildschirm in der Farbe der Variantenattribute dargestellt.

Die Plotsichtbarkeitsmodi werden variantenabhängig gespeichert. Damit ist es möglich, Schaltungsteile abhängig von der aktiven Variante für die Plotausgabe ein- und auszublenden.

6.3.2 Regelsystemanwendungen für den Leiterkartenentwurf

Bauteiltypspezifische Grafik- und Textanzeige

Durch die Zuweisung von Regeln mit dem Prädikat `llnvis` (Logical Library Name Visibility; Beispiele: siehe Regeldefinitionen `layout_llname_*` aus der Regeldefinitionsdatei `layout.rul` im **User Language**-Programmverzeichnis `baeulc`) an auf Dokumentarlagen definierten Flächen und Texten auf Bauteilebene kann eine Darstellung in Abhängigkeit vom Bauteiltyp, d.h. in Abhängigkeit des Bauteilattributs `$llname` (Logical Library Name) erzwungen werden. Hierzu werden im Layout alle Dokumentarlinien, Dokumentarflächen und Dokumentartexte ausgeblendet, deren `llnvis`-Prädikatwert nicht dem `$llname`-Attributwert des Bauteils entsprechen. Damit lassen sich z.B. für einen Gehäusetyp (z.B. SMD `s1206`) unterschiedliche Bestückungsplangrafiken für Kondensatoren (z.B. `$llname c`) und Widerstände (z.B. `$llname r`) definieren.

Platzierungsvorgaben im Layout

Durch die Zuweisung spezieller Regeln an Bauteile, Padstacks und Pads können für diese Elemente Platzierungsvorgaben wie z.B. Drehwinkel oder Spiegelungsmodi definiert werden. Diese Vorgaben werden von den entsprechenden Funktionen zum manuellen bzw. automatischen Platzieren im **Layouteditor** und im **Autorouter** entsprechend berücksichtigt. Die Regeln sind in der Datei `partplc.rul` im **User Language**-Verzeichnis (`baeulc`) abgelegt und mit `rot0`, `rot90`, `rot180`, `rot270`, `mirroroff` und `mirroron` bezeichnet. `partplc.rul` ist mit dem Rule System Compiler **RULECOMP** zu kompilieren. Anschließend können die Regeln auf Bauteil-, Padstack- oder Padebene mit dem **User Language**-Programm **LDEFMANG** zugewiesen werden.

Leiterbahndarstellung beim manuellen Routing

Über das Regelsystem ist es möglich, die Darstellung von Leiterbahnsegmenten während des manuellen Routings zu beeinflussen. Hierzu wird jeweils geprüft, ob eine der Regeln `lay_edit_wide_filled`, `lay_edit_wide_outline` oder `lay_edit_wide_filldist` (siehe Regeldefinitionsdatei `layout.rul` im **User Language**-Programmverzeichnis) an das aktuelle Layout zugewiesen ist. Mit der Funktion [Editierdarstellung] des **User Language**-Programms **GEDTRACE** kann der gewünschte Darstellungsmodus durch Wahl einer der Optionen [Strichanzeige] (Liniendarstellung, Standardeinstellung), [Füllanzeige] (gefüllte Leiterbahnsegmentdarstellung in tatsächlicher Breite), [Randanzeige] (Umrandungsdarstellung in tatsächlicher Breite), [Füllen & Distanz] (gefüllte Leiterbahnsegmentdarstellung in tatsächlicher Breite mit zusätzlicher Liniendarstellung der netzspezifischen Mindestabstände) oder [Füllen & DRC] (gefüllte Leiterbahnsegmentdarstellung in tatsächlicher Breite mit zusätzlicher Liniendarstellung der netzspezifischen Mindestabstände und DRC) selektiert werden.

Die Darstellung der editierten Bahnsegmente erfolgt mit [Füllen & DRC] wie bei der Option [Füllen & Distanz], es wird jedoch zusätzlich ein DRC durchgeführt. Verletzt ein Bahnsegment die Designregeln, dann ändert sich die Farbe der Distanzlinie von der Leiterbahnfarbe auf weiss. Elemente des aktuell bearbeiteten Leiterbahnnetzes sind vom dieser Prüfung ausgenommen. Es ist zu beachten, dass die dargestellte Distanzlinie den DRC-Abstand von Leiterbahn zu Leiterbahn darstellt. Es ist also unter Umständen möglich, mit der Distanzlinie ein Pad anzuschneiden, ohne dass ein DRC-Fehler angezeigt wird, wenn der Abstand Leiterbahn zu Kupfer geringer eingstellt ist, als der Abstand Leiterbahn zu Leiterbahn.

GEDTRACE kann entweder über [Anwenderfunktion] oder (sofern das **User Language**-Programm **UIFSETUP** aktiviert wurde) über [Weitere Funktionen] im Menü [Leiterbahnen] aufgerufen werden.

Warnung

Der Leiterbahndarstellungsmodus kann nur gesetzt werden, wenn die Regeldefinitionsdatei `layout.rul` aus dem **User Language**-Verzeichnis und damit die Regeln `lay_edit_wide_filled` und `lay_edit_wide_outline` mit dem Rule System Compiler **RULECOMP** kompiliert wurden. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei `brules.vdb` im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner müssen die Regeln auch auf diesem bereitgestellt bzw. kompiliert werden, damit der Modus für die Leiterbahndarstellung gesetzt bzw. berücksichtigt werden kann.

Viaprüfbereich für manuelles Routen

Im **Layouteditor** kann wahlweise eine Prüffunktion zur Anzeige aktuell zulässiger Positionen für Durchkontaktierungen während des manuellen Verlegens von Leiterbahnen aktiviert werden. Hierzu ist mit der Funktion Viacheckbereich aus dem Untermenü Weitere Funktionen des Menüs Leiterbahnen ein Viaprüfbereich in Vielfachen des Eingaberasters anzugeben. Es werden Werte von eins bis fünf zur Auswahl angeboten. Im Viaprüfbereich eins wird lediglich der aktuelle Rasterpunkt geprüft, der Viaprüfbereich zwei umfasst zusätzlich die Rasterpunkte um den aktuellen Rasterpunkt (also insgesamt neun Rasterpukte), usw. Mögliche bzw. zulässige Positionen zum Setzen von Durchkontaktierungen innerhalb des gewählten Viaprüfbreichs werden durch kleine weisse Kreise gekennzeichnet. Der Durchmesser dieser Kreise entspricht der Breite der aktuell bearbeiteten Leiterbahn, maximal jedoch 40 Prozent der Eingaberasterschrittweite. Bei freigegebenem Raster oder Auswahl der Viacheckbereich-Option Kein Viacheck ist der Viacheck inaktiv.

Darstellung gestrichelter Linien

Über das Regelsystem ist es möglich, die Darstellung gestrichelter Dokumentarlinien im Layout zu beeinflussen. Hierzu wird jeweils geprüft, ob eine der Regeln **`poly_dash1`**, **`poly_dash2`** oder **`poly_dash3`** (siehe Regeldefinitionsdatei **`polygon.rul`** im **User Language**-Programmverzeichnis) an die Dokumentarlinie zugewiesen ist. **`poly_dash1`** entspricht der Standardeinstellung zur normalen Darstellung gestrichelter Linien. **`poly_dash2`** erzeugt kurze Striche getrennt durch lange Lücken. **`poly_dash3`** bewirkt die Darstellung gestrichelter Linien durch abwechselnd kurze und lange Striche, getrennt durch kurze Lücken. Die Zuweisung dieser Regeln an Dokumentarlinien kann mit Hilfe des **User Language**-Programms **GEDPOLY** über die Option Strichelung durchgefürt werden. Hierbei besteht auch die Möglichkeit, einen eventuell eingestellten Strichelungsmodus mit Durchgezogen wieder zu deaktivieren. **GEDPOLY** kann entweder über Anwenderfunktion oder (sofern das **User Language**-Programm **UIFSETUP** aktiviert wurde) über Weitere Funktionen im Menü Flächen aufgerufen werden.

Warnung

Die Zuweisung von Strichelungsmodi an Dokumentarlinien ist nur möglich, wenn die Regeldefinitionsdatei **`polygon.rul`** aus dem **User Language**-Verzeichnis und damit die Regeln **`poly_dash1`**, **`poly_dash2`** und **`poly_dash3`** mit dem Rule System Compiler **RULECOMP** kompiliert wurden. Es ist weiterhin zu beachten, dass **RULECOMP** die kompilierten Regeln in der Datei **`brules.vdb`** im BAE-Programmverzeichnis ablegt. Bei Übertragung des Designs auf einen anderen Rechner müssen die Regeln auch auf diesem bereitgestellt bzw. kompiliert werden, damit der Modus für gestrichelte Dokumentarlinien gesetzt bzw. berücksichtigt werden kann.

Polygonlinienbreite

Mit der Funktion Polygonlinienbreite setzen des Untermenüs Weitere Funktionen aus dem Menü Flaechen können im **Layouteditor** indivduelle Linienbreiten an einzelne Dokumentarlinien und Split-Power-Plane-Flächen zugewiesen werden. Diese Linienbreiten werden bei der Bilddarstellung und der Plotausgabe entsprechend berücksichtigt. Beim Setzen der Strichstärke auf 0.0 werden die Polygonlinien wie bisher behandelt, d.h. auf dem Bildschirm werden Linien mit einem Pixel Breite dargestellt und bei der Plotausgabe wird die Standardlinienbreite für die Polygonlinien verwendet.

Die eingestellte Linienbreite fuer Split-Power-Plane-Flächen wird auch von der Connectivity berücksichtigt. Berührt eine Bohrung die Breite einer Split-Power-Plane-Linie, so wird die Bohrung auf jeden Fall auf der entsprechenden Versorgungslage abisoliert.

Textstrichstärke

Mit der Funktion Text Stiftbreite setzen des Untermenüs Weitere Funktionen aus dem Menü Text können im **Layouteditor** individuelle Strichstärken an selektierbare Texte zugewiesen werden. Diese werden bei der Bilddarstellung und der Plotausgabe entsprechend berücksichtigt. Beim Setzen der Strichstärke auf 0.0 werden Texte wie bisher behandelt, d.h. auf dem Bildschirm werden Textlinien mit einem Pixel Breite dargestellt und bei der Plotausgabe wird die Standardlinienbreite für die Textlinien verwendet.

Höhen-DRC

Mit der Funktion Hoehen-DRC im Untermenü Weitere Funktionen des **Layouteditor**-Menüs Bauteile können Vorgaben für (Bauteilhöhenrestriktionen definiert werden. Mit der Option Hoehenoffset kann hierzu für selektierbare Bauteile ein Höhenoffset für den Bauteil-DRC vorgegeben werden. Der Bauteilhöhenoffset gibt den vertikalen Abstand des Bauteils von der Leiterkartenoberfläche an und wird bei der Bauteilprüfung zu den Höhenangaben der auf dem Bauteil platzierten Sperrflächen hinzuaddiert. Der durch den Höhenoffset definierte vertikale Bereich zwischen Leiterkarte und Bauteil kann zur Platzierung anderer Bauteile verwendet werden. Über die Option Checkausschluss der Funktion Hoehen-DRC kann ein Alternativbauteil zugewiesen werden, gegen das kein Bauteilhöhen-DRC durchgeführt wird. Damit lassen sich an derselben Leiterkartenposition unterschiedliche Bauteile für eine wahlweise Bestückung platzieren.

Mit den Optionen Hoehenangabe und Hoehenlimit der Funktion Hoehen-DRC aus dem Untermenü Weitere Funktionen des **Layouteditor**-Menüs Flaechen können Höhen bzw. Höhenlimits an Sperrflächen auf Dokumentarlagen zugewiesen werden. Diese Entwurfsvorgaben werden vom DRC entsprechend berücksicht. Üblicherweise wird man auf Bauteilebene Sperrflächen für den vom Bauteil belegten vertikalen Bereich anlegen und mit einer entsprechenden Höhenangabe versehen. Auf Layoutebene werden dann Sperrflächen für in der Höhe beschränkte Bereiche definiert und mit einem Höhenlimit versehen.

Überschneidungen von Sperrflächen mit Höhenangabe werden unabhängig von der Höhenangabe als Dokumentarabstandsfehler behandelt. Sperrflächen mit einem Höhenlimit ungleich 0 werden nicht gegen andere Sperrflächen mit Höhenlimit verglichen.

Höhenfehler werden durch ein Rechteck mit durchgezogenen Diagonalen um den betroffenen Bereich markiert. Zusätzlich wird in dem über die Funktion Report aus dem Menü Utilities abrufbaren Report ggf. der Eintrag `Anzahl Hoehenfehler` zur Anzeige der vom Design Rule Check (DRC) erkannten Höhenfehler angezeigt.

Füllnetzzuweisung an isolierte Vias

Mit der Funktion Fuellnetz setzen aus dem Untermenü Via-Funktionen des Menüs Leiterbahnen kann im **Layouteditor** ein Defaultnetz für das Flächenfüllen an die aktuell zur Gruppe selektierten Durchkontaktierungen zugewiesen werden. Isolierte Vias mit einer Füllnetzzuweisung werden in der Flächenautomatik (siehe Kapitel 4.6.8) als dem zugewiesenen Füllnetz zugehörig betrachtet und bei Übereinstimmung mit dem Füllbereichsnetz an die erzeugten Füllflächen angeschlossen. Damit kann die Generierung von Wärmefallen in netzspezifischen Füllbereichen erzwungen werden. Für Vias mit bestehenden physikalischen Verbindungen zu Signalnetzen ist die Füllnetzvorgabe bedeutungslos.

Bohrungsanbindung an Versorgungslagen

Das Untermenü Weitere Funktionen aus dem Menü Texte, Bohrungen des **Layouteditors** enthält die beiden Funktionen Bohrung V-Lagen und Bohrung Waermefallen zur Zuweisung von Versorgungslagenparametern an mausselektierbare Bohrungen auf Padstackebene. Mit Bohrung V-Lagen kann vorgegeben werden, welche Versorgungslagen von der selektierten Bohrung belegt sind. Bohrung Waermefallen steuert die Anschlussart der selektierten Bohrung für die einzelnen Versorgungslagen. Dabei kann zwischen Wärmefalle (Default) und Direktanschluss gewählt werden. Zur Zuweisung werden jeweils in einer Schleife Bohrungen selektiert und anschließend in einer Dialogbox die Vorgaben für die Versorgungslagenbelegung bzw. die versorgungslagenspezifische Anschlussart gemacht. In Arbeitsumgebungen ohne Unterstützung von Dialogboxen ist jeweils eine Bitmaske mit für die einzelnen Versorgungslagen gesetzten Bits einzugeben.

Spezifische Regelvorgaben für die Anbindung von Bohrungen an Versorgungslagen sind insbesondere für die Definition von partiellen Durchkontaktierungen von Bedeutung und werden bei der Connectivity, bei der Bilddarstellung der Versorgungslagen sowie bei der CAM-Ausgabe entsprechend berücksichtigt.

Kapitel 7 Utilities

Dieses Kapitel beschreibt die Utilityprogramme des **Bartels AutoEngineer**. Diese Programme laufen üblicherweise im Batchbetrieb, d.h. sie können von der Betriebssystemebene aus aufgerufen werden und benötigen kein Grafikinterface für den Dialog mit dem Benutzer. Die Utilityprogramme sind nützliche Zusatzwerkzeuge für die Bibliotheksverwaltung, zum Auswerten von DDB(Design DataBase)-Files, zur Konfiguration des **AutoEngineers**, zum Umsetzen von (Fremd-)Netzlisten, usw. usf.

7.1 BAEHELP

Name

baehelp - BAE Windows Online-Dokumentation

Synopsis

```
baehelp [file]
```

Beschreibung

Das Programm **BAEHELP** aktiviert unter Windows den Default-Webbrowser zum Laden der wahlweise über das Argument **`file`** angegebenen (HTML-)Datei bzw. URL. Ist kein Argument angegeben, dann wird automatisch das BAE Benutzerhandbuch aus dem Verzeichnis **`../baedoc`** relativ zum BAE-Programmverzeichnis geladen.

Warnungen

BAEHELP ist nur unter Windows ablauffähig.

7.2 BAESETUP, BSETUP

Name

baesetup - Bartels AutoEngineer Setup Modul
bsetup - Bartels AutoEngineer Setup Utility

Synopsis

```
bsetup -encode <code>

bsetup setupfile
```

Beschreibung

Freischaltung von BAE-Software-Autorisierungen

Der Aufruf

```
bsetup -encode <code>
```

dient der Freischaltung von BAE-Autorisierungen im Feld. Die Option **`-encode`** erwartet als Argument den Autorisierungscode für die freizuschaltende BAE-Software-Konfiguration. Dieser Autorisierungscode ist im Bedarfsfall bei der Bartels System GmbH erhältlich. Der **BSETUP**-Aufruf muss aus dem aktuellen BAE-Programmverzeichnis auf der Maschine, an der der freizuschaltende Hardlock-Key installiert ist, erfolgen. Unmittelbar nach Aufruf des Utilityprogramms **BSETUP** mit der Option **`-encode`** (und einem *korrekten* Autorisierungscode) ist der **Bartels AutoEngineer** einmal zum Zwecke der Übernahme bzw. Freischaltung der neuen Autorisierung aufzurufen. Erst bei nachfolgenden BAE-Aufrufen erfolgt dann eine korrekte Autorisierungsprüfung, d.h. nach dem BAE-Aufruf zur Freischaltung der neuen Autorisierung (zu beachten ist hierbei die Meldung **`Neue Optionen : <sw-config>.`** in der Statuszeile) muss ein Programmabbruch veranlasst werden bevor ein produktives Arbeiten mit dem **AutoEngineer** wieder möglich ist.

Übernahme und Modifikation von BAE-Setupdaten

Der Aufruf

```
bsetup setupfile
```

dient der Übernahme von Setupdaten in den **Bartels AutoEngineer**, d.h. die anwenderspezifische Gestaltung der BAE-Benutzeroberfläche, die firmenrelevante Definition der Lagenzuordnung in den Layoutmenüs sowie die Einstellung von Standard-Bibliotheks-Suchpfaden und -Namen, usw. Hierbei erwartet **BSETUP** als Argument den Dateinamen **`setupfile`** der Setupdatei (diese Datei muss mit der Extension **`.def`** verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben). **BSETUP** liest die Setupdatei und speichert die darin enthaltenen Setup-Parameter in der Datei **`bsetup.dat`** (im aktuellen Verzeichnis) ab. Beim Aufruf des **AutoEngineers** werden die in dieser Datei gespeicherten Setup-Parameter eingelesen und sind somit für die Dauer der Bearbeitung im **AutoEngineer** verfügbar (Voraussetzung hierfür ist die Installation von **`bsetup.dat`** im BAE-Programm-Verzeichnis).

Unter Windows, Linux und Unix können die nachfolgend aufgeführten Systemparameter wahlweise auch mit dem über die Funktion Setup aus dem BAE-Hauptmenü aufrufbaren **BAESETUP** modifiziert werden. **BAESETUP** ist dialogboxorientiert und daher wesentlich einfacher zu bedienen als **BSETUP** vor dessen Aufruf üblicherweise eine DEF-Datei mit dem kompletten BAE-Parameterdatensatz erstellt bzw. modifiziert werden muss. **BAESETUP** bietet zusätzlich die Möglichkeit, die geänderten Setupdaten in eine **BSETUP**-kompatible DEF-Datei zu exportieren.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Setupdatei beginnt mit dem Schlüsselwort **SETUP** und endet mit **END.**. Kommentare können zwischen **/*** und ***/** eingefügt werden.

Kommando LAYMENUTEXT

Mit dem Kommando **LAYMENUTEXT** lassen sich die Menüs für die Signallagen-Auswahl festlegen. Die Syntax hierfür lautet:

```
LAYMENUTEXT LINE <line> ("<text>",<layer>);
```

Über **<line>** ist die Menüzeile (im Bereich von 1 bis 12) anzugeben. Der Eintrag <text> gibt den auf dem Menüfeld erscheinenden Text an und kann vom Benutzer frei vergeben werden. **<layer>** gibt die durch das Menüfeld selektierte (Signal-)Lage an, wobei von 1 bis 100 gewählt werden kann. Eine Sonderform des **LAYMENUTEXT**-Kommandos ist

```
LAYMENUTEXT TOPLAYER ("<text>");
```

Obige Zeile, in der lediglich **<text>** geändert werden darf, ermöglicht die Definition einer Signallage als Oberste Lage.

Kommando LAYPADLAYER

Mit dem Kommando **LAYPADLAYER** kann eine Lagenzuordnung auf Padebene ermöglicht (**ENABLE**) oder verhindert werden (**DISABLE**). Die Syntax für dieses Kommando lautet:

```
LAYPADLAYER (ENABLE) ;
```

bzw.

```
LAYPADLAYER (DISABLE) ;
```

Dieses Kommando ist historisch bedingt, da in früheren Software-Versionen auf Padebene die Zuweisung von Flächen auf Lagen möglich war. Der Eintrag **ENABLE** sollte also nurmehr dazu dienen, alte Job-Files zu aktualisieren. Wir empfehlen ansonsten den Eintrag **DISABLE** (siehe **stdset.def**).

Kommando LAYPLTMARKLAY

Über das Kommando **LAYPLTMARKLAY** lässt sich eine Dokumentarlage definieren die durch den **CAM-Prozessor** im Alle Lagen-Modus grundsätzlich mit ausgegeben wird (sinnvoll ist hier z.B. die Lage für die Film-Passermarken). Die Syntax hierfür lautet:

```
LAYPLTMARKLAY (<layer>) ;
```

Kommando LAYGRPDISPLAY

Mit dem Kommando **LAYGRPDISPLAY** lässt sich eine Dokumentarlage bzw. Darstellungsebene definieren, die mit der **Layouteditor**-Gruppenfunktion Bewegtdarstellung über den Menüpunkt Nur Baugruppenlage ausschließlich zur Darstellung bewegter Gruppen angezeigt werden soll. Die Syntax für dieses Kommando lautet:

```
LAYGRPDISPLAY (<layer>) ;
```

Kommando LAYDOCLAYER

Mit dem Kommando **LAYDOCLAYER** können bis zu 100 Dokumentarlagen definiert werden. Dabei ist zu beachten, dass eine Dokumentarlage im Grunde aus 3 "Seiten", nämlich Seite 1 (unterste Lage, Lötseite), Seite 2 (oberste Lage, Bauteilseite) und Beide Seiten bestehen kann. Damit ist es möglich, Elemente der Seite 1 der Dokumentarlage auf die Seite 2 zu spiegeln und umgekehrt (z.B. SMD-Bauteil von Bauteilseite auf Lötseite spiegeln mit entsprechender Spiegelung von am SMD-Bauteil definierten Dokumentar-Grafik, Dokumentar-Text, ...). Beim Plotten werden - sofern der Modus Alle Lagen eingeschaltet ist - die unter Beide Seiten abgelegten Elemente sowohl beim Plotten der Seite 1, als auch beim Plotten der Seite 2 hinzugefügt. Die Syntax für das **LAYDOCLAYER**-Kommando lautet:

```
LAYDOCLAYER <layer> ("<text>",<side>,<rotate>[,<index>]) ;
```

Der Eintrag **<layer>** gibt die Nummer der Dokumentarlage an, wobei von 1 bis 100 gewählt werden kann. Der Eintrag **<text>** gibt den vom Benutzer frei wählbaren Text an, der auf dem Menüfeld als Bezeichnung für die Dokumentarlage erscheinen soll. **<side>** gibt den Modus für die Abfrage nach der Seite bei der Generierung von Elementen auf der entsprechenden Dokumentarlage an, wobei folgende Einträge erlaubt sind:

SIDE1	Eingaben erfolgen automatisch auf Seite 1 der Dokumentarlage
SIDE2	Eingaben erfolgen automatisch auf Seite 2 der Dokumentarlage
BOTH	Eingaben erfolgen automatisch auf Beide Seiten der Dokumentarlage
NONE	Eingaben auf der Dokumentarlage mit Möglichkeit der Auswahl der Dokumentarlagenseite (Optionen Seite 1, Seite 2, Beide Seiten)

Der Parameter **<rotate>** erlaubt die Festlegung der Richtlinien für das Spiegeln und Rotieren von Texten, wobei folgende Einträge erlaubt sind:

LOGICAL	Text kann frei gedreht und gespiegelt werden und bleibt dabei leserichtig (z.B. fü Bestückungsplan)
PHYSICAL	erlaubt beliebiges Drehen und Spiegeln des Textes, wobei jedoch dessen Position nicht durch die Funktionen Name bewegen bzw. Attribut bewegen verändert werden kann (z.B. für Bestückdaten)
NOROTATE	Bauteil kann frei gedreht und gespiegelt werden, der Text bleibt jedoch fixiert (ansonsten wie **PHYSICAL**; z.B. für Bohrplan)

Über den Parameter **<index>** kann optional ein Ausgabeindex für die Festlegung der Anzeigereihenfolge der Dokumentarlagen in der Farbpalette und in den Lagenauswahlmenüs vorgegeben werden. Die Indexnummerierung beginnend bei 1. Dokumentarlagen ohne Indexvorgabe werden fortlaufend auf die noch freien Anzeigepositionen verteilt. Mit Hilfe des **<index>**-Parameters können häufig benutzte Dokumentarlagen am Beginn der Dokumentarlagenauswahl platziert werden, und die Dokumentarlagen lassen sich in der Dokumentarlagenauswahl entsprechend ihrer Funktionalität gruppieren.

Kommando DOCMENU

Mit dem Kommando **DOCMENU** können Dokumentarlagen an die oberste Ebene in den Lagenauswahlmenüs des Layoutsystems zugewiesen werden um die Auswahl häufig benutzter Dokumentarlagen zu vereinfachen bzw. zu beschleunigen. Die Syntax für das **DOCMENU**-Kommando lautet:

```
DOCMENU <menuline> (<layer>) ;
```

<menuline> ist hierbei die Zeile bzw. Position an der die über die Dokumentarlagennummer **<menuline>** angegebene Dokumentarlage in den obersten Ebenen der Lagenauswahlmenüs angezeigt werden soll.

Kommando USERUNITS

Mit dem Kommando **USERUNITS** kann die Voreinstellung der Längeneinheiten für die Koordinatenein- und ausgabe definiert werden. Die Syntax für dieses Kommando lautet:

```
USERUNITS (METRIC) ;
```

bzw.

```
USERUNITS (IMPERIAL) ;
```

Dabei gibt **METRIC** eine Einstellung auf mm-Einheiten und **IMPERIAL** eine Einstellung auf Inch-Einheiten an.

Kommando SCMDEFLIBRARY

Mit dem Kommando **SCMDEFLIBRARY** lässt sich ein Verweis auf die Stromlaufbibliothek **<stdlib>.ddb** mit Standardsymbolen (Pin, Bustap, Junction, Labels, ...) und gleichzeitig ein Verweis auf das Verzeichnis **<libpath>** mit den Stromlaufbibliotheken einstellen. Die Syntax hierfür lautet:

```
SCMDEFLIBRARY ("<libpath>/<stdlib>");
```

Der Datenbank-Pfad **<libpath>** ist dabei frei wählbar, sollte jedoch auf das Verzeichnis verweisen, in dem neben der Datei **<stdlib>.ddb** mit den Standardsymbolen auch alle anderen benötigten Stromlauf-Bibliotheksdateien zu finden sind. Zu beachten ist dabei, dass **/** das Trennzeichen für Hierarchien (sowohl im Dateiverzeichnis, als auch auf Datenbank-Ebene) darstellt (im Unterschied zu DOS: ****). Bei richtiger Definition dieser System-Variablen kann jedes SCM-Bibliothekssymbol entweder über Popupmenü durch die Selektion der SCM-Bibliotheksdatei und die Anwahl des gewünschten Symbols oder direkt durch die Eingabe von

```
<scmlib>/<symbolname>
```

in den Stromlauf geladen werden (also z.B. mit **74ls/74ls90** oder **passiv/r**). Auf die in **<stdlib>.ddb** definierten Sondersymbole kann ohne explizite Selektion der Bibliotheksdatei direkt zugegriffen werden. Darüber hinaus ist im **Schaltplaneditor** auch weiterhin die explizite Selektion der Stromlaufbibliothek über die Funktion Bibliotheksname im Menü Einstellungen möglich (die Eingabe von **-** bewirkt dabei das Löschen des aktuellen Eintrags, die Eingabe von **!** bzw. **.** die Einstellung der über **SCMDEFLIBRARY** definierten Bibliothek).

Kommando LAYDEFLIBRARY

Mit dem Kommando **LAYDEFLIBRARY** lässt sich der Pfadname der Bibliothek mit den Layoutsymbolen definieren. Die Syntax für dieses Kommando lautet:

```
LAYDDEFLIBRARY ("<libpath>");
```

Der Datenbank-Pfad **<libpath>** ist dabei frei wählbar; zu beachten ist dabei, dass **/** das Trennzeichen für Hierarchien im Dateiverzeichnis darstellt (im Unterschied zu DOS: ****). Bei richtiger Definition dieser System-Variablen kann jedes in der Datei **<libpath>.ddb** enthaltene Layoutsymbol durch die Angabe von

```
<symbolname>
```

in das Layout geladen werden (also z.B. mit **dil16** oder **sot23**). Darüber hinaus ist sowohl im **Layouteditor** als auch im **Autoplacement** auch weiterhin die explizite Selektion der Layoutbibliothek über die Funktion Bibliotheksname im Menü Einstellungen möglich (die Eingabe von **-** bewirkt dabei das Löschen des aktuellen Eintrags, die Eingabe von **!** bzw. **.** die Einstellung der über **LAYDEFLIBRARY** definierten Bibliothek).

Neben dem Default-Pfadnamen für die Symbolbibliothek im BAE-Layoutsystem setzt das Kommando **LAYDEFLIBRARY** zusätzlich auch den Pfadnamen der Standardbibliothek für die Funktion Symbollogik anzeigen zur Anzeige Logischer Bibliotheksdefinitionen im **Schaltplaneditor**.

Kommando LAYDEFELEMENT

Mit dem Kommando **LAYDEFELEMENT** lässt sich ein Default-Eintrag für den Namen des jeweils zu generierenden Layouts festlegen. Die Syntax für dieses Kommando lautet:

```
LAYDEFELEMENT ("<layout-elementname>");
```

Die Definition dieses Eintrages erleichtert das Arbeiten mit dem **Packager** ganz erheblich, da dabei jeweils die Abfrage nach dem Layoutelement einfach mit der Eingabetaste quittiert werden kann, und die Software dann selbsttätig den über **LAYDEFELEMENT** definierten Namen einträgt (selbstverständlich ist auch weiterhin eine explizite Spezifikation des Layoutelementnamens möglich).

Kommando PROJROOTDIR

Mit dem Kommando **PROJROOTDIR** kann das Startverzeichnis für die optionale Directory-Auswahl in den Datei-Auswahlmenüs festgelegt werden. Bei der Verzeichnisauswahl werden dann jeweils die Unterverzeichnisse des mit **PROJROOTDIR** definierten Wurzelverzeichnisses zur Auswahl angezeigt. Die Syntax für das Kommando **PROJROOTDIR** lautet

```
PROJROOTDIR ("<rootdir>");
```

wobei für **<rootdir>** das gewünschte Wurzelverzeichnis für die Directory-Auswahl anzugeben ist. Enthält die Setupdatei kein **PROJROOTDIR**-Kommando, dann wird das aktuelle Verzeichnis für die Verzeichnisauswahl verwendet. Mögliche Einträge für <rootdir> sind z.B. **/** (Wurzelverzeichnis des aktuellen Laufwerks), **d:** (Wurzelverzeichnis des PC-Laufwerks D:), **c:/cad_data** (Verzeichnis **cad_data** auf dem PC-Laufwerk C:), **/pcb/projects** (Verzeichnisbaum **pcb/projects** des aktuellen Laufwerks), usw. Der **<rootdir>**-Eintrag des **PROJROOTDIR**-Kommandos darf keine Sonderzeichen wie z.B. **.** oder **** enthalten.

Kommando WINMENUMODE

Die Windows- bzw. Motif-Versionen der BAE-Software können wahlweise mit Pulldownmenüs oder mit Seitenmenüs betrieben werden. Das **BSETUP**-Kommando **WINMENUMODE** dient dazu, die gewünschte Benutzeroberfläche zu aktivieren. Mit dem folgenden Kommando wird die BAE-Standardbenutzeroberfläche mit Seitenmenüs aktiviert; dies ist zugleich die Standardeinstellung für den Fall, dass in der Setupdatei kein **WINMENUMODE**-Kommando eingetragen ist:

```
WINMENUMODE (SIDEMENU);
```

Mit dem folgenden Kommando wird in den Windows- bzw. Motif-Versionen der BAE-Software die BAE-Benutzeroberfläche mit Pulldownmenüs (Kontextmenüs erreichbar über linke Maustaste, Funktionswiederholung über rechte Maustaste) aktiviert:

```
WINMENUMODE (PULLDOWN);
```

Mit dem folgenden Kommando wird in den Windows- bzw. Motif-Versionen der BAE-Software die BAE-Benutzeroberfläche mit Pulldownmenüs (Kontextmenüs Windows-konform erreichbar über rechte Maustaste, Funktionswiederholung über linke Maustaste) aktiviert:

```
WINMENUMODE (PULLDOWN_RMB_CONTEXT);
```

Die DOS- und X11-Versionen der BAE-Software können grundsätzlich nur mit der BAE-Standardbenutzeroberfläche betrieben werden.

Kommando FRAMECOLOR

Mit dem Kommando **FRAMECOLOR** kann die Farbzuordnung für die Benutzeroberfläche der BAE-Module definiert werden. Die Syntax für dieses Kommando lautet:

```
FRAMECOLOR <screenarea> (<colornumber>);
```

<screenarea> gibt den Arbeitsbereich der Benutzeroberfläche an. Hierbei sind folgende Einträge für die Benutzeroberflächen der BAE-Grafikmodule (BAE-Shell, **Schaltplaneditor**, **Layouteditor**, **Autorouter**, **CAM-Prozessor**, **CAM-View**) möglich:

Identifier	Arbeitsbereich
DIALAREA	Status-/Eingabezeile
LISTAREA	Textausgabe/Grafikarbeitsbereich
MENUHEAD	Menü-Header/Info-Feld
MENUHEAD BACK	Menü-Header/Info-Feld Hintergrund
MAINMENU	Hauptmenü
MAINMENU BACK	Hauptmenü Hintergrund
SUBMENUA	Menü/Untermenü
SUBMENUA BACK	Menü/Untermenü Hintergrund
EMENMARK	Menü-Kursor enabled (System erwartet Eingabe)
EFILMARK	Menü-Balken enabled (System erwartet Eingabe)
DMENMARK	Menü-Kursor disabled (System arbeitet)
DFILMARK	Menü-Balken disabled (System arbeitet)
POPMTEXT	Popupmenü Text
POPMBUTT	Popupmenü Button
POPMBACK	Popupmenü Hintergrund
POPMFRAM	Popupmenü Rahmen
POPMFILL	Directory-Popupmenü Hintergrund

`<colornumber>` gibt die Farbnummer an. Dabei sind Werteinträge von 1 bis 15 erlaubt. Die Zuordnung der Farbnummer zu der entsprechenden Farbe erfolgt gemäß folgender Liste:

Farbnummer	Farbe
1	Blau
2	Grün
3	Kobaltblau
4	Rot
5	Violett
6	Braun
7	Hellgrau
8	Dunkelgrau
9	Hellblau
10	Hellgrün
11	Hellkobaltblau
12	Hellrot
13	Hellviolett
14	Gelb
15	Weiß

Beispiele

Im Lieferumfang des AutoEngineers ist ein Setup-File-Template (nach Installation im BAE-Programm-Verzeichnis unter **stdset.def**) mit folgendem Inhalt enthalten:

```
SETUP


/* Bartels AutoEngineer Standard Setup */


/* Menue Lagenbezeichnungen */
LAYMENUTEXT LINE 1   ("Lage &1 (Loets.)",1);
LAYMENUTEXT LINE 2   ("Lage &2",2);
LAYMENUTEXT LINE 3   ("Lage &3",3);
LAYMENUTEXT LINE 4   ("Lage &4",4);
LAYMENUTEXT TOPLAYER ("Lage n (Bes&ts.)");


/* Dokumentarlagen */
LAYDOCLAYER 1 ("Bestueckungsplan",SIDE2,LOGICAL);
LAYDOCLAYER 2 ("Loetmaske",NONE,PHYSICAL);
LAYDOCLAYER 3 ("Bohrplan",BOTH,NOROTATE);
LAYDOCLAYER 4 ("Film Passermarken",BOTH,PHYSICAL);
LAYDOCLAYER 5 ("Baugruppen",BOTH,LOGICAL);
LAYDOCLAYER 6 ("Bauteil-DRC",SIDE2,LOGICAL);
LAYDOCLAYER 7 ("Pin-Nummer",SIDE2,LOGICAL);
LAYDOCLAYER 8 ("Lotauftrag (SMD)",SIDE2,LOGICAL);
LAYDOCLAYER 9 ("Bemassung/Hinweise",SIDE2,LOGICAL);


/* Pad Lagenabfrage Freigabe */
LAYPADLAYER (DISABLE);
```

```
/* Film Passermarken */
LAYPLTMARKLAY (4);

/* Teilweise Gruppen Darstellung */
LAYGRPDISPLAY (5);

/* Standard Suchpfade und Namen */
SCMDEFLIBRARY ("/baelib/stdsym");
LAYDEFLIBRARY ("/baelib/laylib");
LAYDEFELEMENT ("s1");

/* Standard Benutzereinheiten */
USERUNITS (METRIC);

/* Windows/Motif Menueform */
WINMENUMODE (PULLDOWN);

/* Farbeinstellungen Grafik allgemein */
FRAMECOLOR DIALAREA (11);
FRAMECOLOR LISTAREA (14);

/* Farbeinstellungen Seitenmenu rechts */
FRAMECOLOR MENUHEAD (10);
FRAMECOLOR MENUHEAD BACK (8);
FRAMECOLOR MAINMENU (12);
FRAMECOLOR MAINMENU BACK (8);
FRAMECOLOR SUBMENUA (10);
FRAMECOLOR SUBMENUA BACK (8);
```

```
/* Farbeinstellungen Selektionsbalken */

FRAMECOLOR EMENMARK (2);

FRAMECOLOR DMENMARK (15);

FRAMECOLOR EFILMARK (8);

FRAMECOLOR DFILMARK (4);



/* Farbeinstellungen Popupmenu */

FRAMECOLOR POPMTEXT (3);

FRAMECOLOR POPMBUTT (14);

FRAMECOLOR POPMBACK (8);

FRAMECOLOR POPMFRAM (15);

FRAMECOLOR POPMFILL (1);



/* Farbeinstellungen Textprogramme */

FRAMECOLOR DIALLINE (10);

FRAMECOLOR OUTLINES (14);

FRAMECOLOR HEADLINE (12);


END.
```

Obige Setupdatei sollte an die benutzer- bzw. firmenspezifischen Anforderungen (Aufbau der Benutzeroberfläche, Zugriff auf Stromlauf- und Layoutbibliotheken, Zuordnung der Signallagen, Verwendung von Dokumentarlagen, usw.) angepasst werden und kann dann mit dem folgenden Kommando in die Software eingespielt werden:

```
> bsetup stdset ↵
```

Dateien

bsetup.dat -- Setupdatei kompiliert (im BAE-Programm-Verzeichnis)
stdset.def -- Setupdatei-Quellcode (Template)

Siehe auch

BAE Shell, **Schaltplaneditor**, **Layouteditor**, **Autorouter**, **CAM-Prozessor**, **CAM-View**, **Packager**.

Diagnose

Die durch **BSETUP** erzeugten Fehlermeldungen sind selbsterklärend.

7.3 BICSET (IC-Design)

HINWEIS: Das Utilityprogramm **BICSET** ist nur in **Bartels AutoEngineer IC-Design** verfügbar!

Name

bicset - Bartels AutoEngineer IC-Design Setup Utility

Synopsis

```
bicset setupfile
```

Beschreibung

Das Utilityprogram **BICSET** dient der Konfiguration von Setupdaten für das IC-Designsystem des **Bartels AutoEngineer IC-Design**-System. Hierzu zählen Standardzellen-Dimensionsangaben für die automatische Platzierung, Lagenzuordnungen und Lagenmenüdefinitionen, DRC-Parameter, usw. **BICSET** erwartet als Argument den Dateinamen **`setupfile`** der Setupdatei (diese Datei muss mit der Extension **`.def`** verfügbar sein; beim Programmaufruf ist diese Dateinamensendung wegzulassen). **BICSET** liest die Setupdatei und speichert die darin enthaltenen Setup-Parameter in der Datei **`bsetup.dat`** (im aktuellen Verzeichnis) ab. Beim Aufruf des IC-Designsystems des **Bartels AutoEngineers IC-Design**-Systems werden diese Parameter aus der Datei **`bsetup.dat`** im BAE-Programm-Verzeichnis geladen und aktiviert.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Setupdatei beginnt mit dem Schlüsselwort **`SETUP`** und endet mit **`END.`**. Kommentare können zwischen **`/*`** und **`*/`** eingefügt werden.

Dieser Abschnitt wird zur Zeit überarbeitet. Wir bitten um Nachsicht.

Beispiele

Im Lieferumfang der **Bartels AutoEngineer IC-Design** Software ist die IC-Design-Setupdatei `icset.def` (im BAE-Programm-Verzeichnis) mit folgendem Inhalt enthalten:

```
SETUP


/* Bartels AutoEngineer IC Design Setup */

```

```
Dieser Abschnitt wird zur Zeit überarbeitet. Wir bitten um Nachsicht.
```

```


END.
```

Obige Setupdatei können Sie an Ihre spezifischen Anforderungen anpassen und dann mit dem folgenden Kommando kompilieren bzw. aktivieren:

```
> bicset icset ↵
```

Dateien

`bsetup.dat` -- Setupdatei kompiliert (im BAE-Programm-Verzeichnis)
`icset.def` -- Setupdatei-Quellcode (Template)

Siehe auch

BAE-Shell, **Chipeditor**, **Cellplacer**, **Cellrouter**.

Diagnose

Die durch **BICSET** erzeugten Fehlermeldungen sind selbsterklärend.

7.4 BLDRING (IC-Design)

HINWEIS: Das Utilityprogramm **BLDRING** ist nur in **Bartels AutoEngineer IC-Design** verfügbar!

Name

bldring - Bartels AutoEngineer IC-Design Build Ring Utility

Synopsis

```
bldring ringdescriptionfile
```

Beschreibung

Das Utilityprogram **BLDRING** dient zur automatisierten Erstellung eines Chip-Basislayouts aus einer Beschreibungsdatei. Die Beschreibungsdatei entählt Informationen über die verwendeten Zellmakros und Sektionen zur Platzierung eines ässeren (rechteckigen) Rings mit Bondingpadzellen und beliebigen anderen Zellplatzierungen. Bei den Bondingpadzellen muss nur die relative Abfolge der Padzellen in den einzelnen Ringseiten spezifiziert werden, die Platzierungskoordinaten werden automatisch aus den Zellmakrobeschreibungen und den Ringdimensionen bestimmt. In den nicht von Bondingpads belegten Teilen des Rings werden automatisch Metallstrukturen zum Durchschleifen zweier Spannungsversorgungen erzeugt. Bei den festen Zellplatzierungen ist eine Koordinatenangabe relativ zu den Begrenzungen des Rings möglich, so dass einmal definierte Strukturen einfach in verschieden grosse Designs übernommen werden können. **BLDRING** erwartet als Argument den Dateinamen `ringdescriptionfile` der Beschreibungsdatei (diese Datei muss mit der Extension `.rig` verfügbar sein; beim Programmaufruf ist diese Dateinamensendung wegzulassen). Die Namen der Ausgabedatei und des darin erzeugten IC-Designelementes werden in der Ringbeschreibungsdatei spezifiziert.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Setupdatei beginnt mit dem Schlüsselwort `ring` und wird durch das Dateiende beendet. Kommentare können zwischen `/*` und `*/` eingefügt werden.

Dieser Abschnitt wird zur Zeit überarbeitet. Wir bitten um Nachsicht.

Beispiele

Dieser Abschnitt wird zur Zeit überarbeitet. Wir bitten um Nachsicht.

Dateien

Siehe auch

Chipeditor.

Diagnose

Die durch **BLDRING** erzeugten Fehlermeldungen sind selbsterklärend.

7.5 CONCONV

Name

conconv - Connections Conversion Utility

Synopsis

```
conconv projectname libraryfile
```

Beschreibung

Das Programm **CONCONV** dient der Übertragung von physikalischen (d.h. gepackten) ASCII-Netzlisten aus den Formaten BAE, CALAY, MARCONI oder RACAL in das interne Netzlistenformat des **Bartels AutoEngineer**.

CONCONV erwartet als erstes Argument den Dateinamen **`projectname`** der Netzlistendatei (diese Datei muss mit der Extension **`.con`** verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben).

CONCONV erwartet als zweites Argument den Namen **`libraryfile`** der Layoutbibliothek, aus der die Gehäusebauformen geladen werden sollen. **`libraryfile`** muss im DDB(Design DataBase)-Format vorliegen, und im Dateinamen die Extension **`.ddb`** aufweisen (diese Extension ist beim Programmaufruf nicht mit anzugeben).

CONCONV liest die ASCII-Netzliste **`<projectname>.con`** und prüft, ob alle darin enthaltenen Layoutsymbole in der Bibliothek **`<libraryfile>.ddb`** definiert sind. Es wird das Job-File **`<projectname>.ddb`** generiert und die Freelist **`<projectname>.fre`** ausgegeben. Die Freelist wird vom System nicht weiter benötigt und ist zur Auswertung durch den Benutzer bestimmt (Anzeige nicht angeschlossener Pins, Statistik). Im Job-File befindet sich nunmehr die übersetzte Netzliste. Auch die ASCII-Netzliste **`<projektname>.con`** wird jetzt nicht mehr benötigt, sollte jedoch für etwaige Änderungen aufgehoben werden.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Eingabedatei mit der einzulesenden ASCII-Netzliste muss mit dem Kommando **`LAYOUT`** beginnen und mit dem Kommando **`END.`** enden. Kommentare können an beliebiger Stelle zwischen **`/*`** und **`*/`** eingefügt werden. Das **`LAYOUT`**-Kommando hat folgendes Format:

```
LAYOUT <elementname>;
```

<elementname> gibt den Namen der Netzliste bzw. des in der Zieldatei zu erstellenden Layouts an.

Bauteilliste

Nach dem **`LAYOUT`**-Kommando folgt die Bauteilliste, die mit dem Schlüsselwort **`PARTS`** beginnen muss und für jedes Bauteil ein Kommando der Form

```
<part> : <plname> ;
```

enthält. <part> ist hierbei der Bauteilname, <plname> der Physical Library Name, d.h. der Name des Gehäuses bzw. des Layoutsymbols.

Netzliste

Nach der Bauteilliste folgt die Netzliste, die je nach Format mit einem entsprechenden Schlüsselwort beginnen muss (**CONNECT** für Bartels-Format, **CALAY** für CALAY-Format, **RACAL** für RACAL-Format, **MARCONI** für MARCONI-Format). Im Bartels-Format ist für jedes Netz ein Kommando der Form

```
<part>.<pin>=<part>.<pin>=...=<part>.<pin>
```

bzw.

```
/<net>/ <part>.<pin>=<part>.<pin>=...=<part>.<pin>
```

einzutragen. Zusätzlich können hinter **"/<net>/"** die folgenden Netzattribute definiert werden:

```
PRIORITY(<prior>) MINDIST(<dist>) ROUTWIDTH(<width>)
```

Obige Netzattribute werden vom **Autorouter** berücksichtigt, wobei **<prior>** die Routingpriorität, **<dist>** den minimalen Abstand des Netzes zu anderen Potentialen und **<width>** die netzbezogene Leiterbahnbreite spezifizieren. Darüber hinaus kann für jeden einzelnen Pin (jeweils hinter **"<pin>"**, in Klammern) eine pinbezogene Leiterbahnanschlussbreite definiert werden. **<prior>** ist als Integerwert, die anderen Werte jeweils in Millimetern anzugeben.

Im CALAY-Format ist für jedes Netz ein Kommando der Form

```
<part>(<pin>),<part>(<pin>),...,<part>(<pin>)
```

bzw.

```
/<net> <part>(<pin>),<part>(<pin>),...,<part>(<pin>)
```

einzutragen. Darüber hinaus kann für jeden einzelnen Pin (jeweils hinter **"<pin>"**, abgetrennt durch ein Komma) eine pinbezogene Leiterbahnanschlussbreite (Einheit mm) definiert werden.

Im RACAL-Format ist für jedes Netz eine Kommandosequenz der Form

```
.ADD_TER        <part> <pin> <net>

.TER            <part> <pin>

                <part> <pin>

                :

                <part> <pin>
```

einzutragen. Die RACAL-Netzliste ist mit dem Schlüsselwort **.END** abzuschließen.

Im MARCONI-Format ist für jedes Netz ein Kommando der Form

```
<part> <pin> <part> <pin> ... <part> <pin> ; <net> /
```

einzutragen.

Beispiele

Netzliste (`design.con`) im Bartels-Format:

```
LAYOUT board;

PARTS

        c1 : cap50;

        c2 : cap75;

        r1 : res;

        t1 : tebc;

CONNECT

        /net1/ c2.2=t1.3;

        /net2/ c1.2(0.4)=t1.2=r1.2;

        /gnd/ PRIORITY(2) MINDIST(0.4) t1.1=c1.1(0.4);

        /vcc/ PRIORITY(1) ROUTWIDTH(0.5) c2.1=r1.1;

END.
```

Netzliste (`design.con`) im CALAY-Format:

```
LAYOUT board;

PARTS

        c1 : cap50;

        c2 : cap75;

        r1 : res;

        t1 : tebc;

CALAY

        /net1 c2(2),t1(3);

        /net2 c1(2,0.4),t1(2),r1(2);

        /gnd t1(1),c1(1,0.4);

        /vcc c2(1,0.5),r1(1,0.5);

END.
```

Netzliste (**design.con**) im RACAL-Format:

```
LAYOUT board;

PARTS

        c1 : cap50;

        c2 : cap75;

        r1 : res;

        t1 : tebc;

RACAL

        .ADD_TER        c2 2 net1

        .TER            t1 3

        .ADD_TER        c1 2 net2

        .TER            t1 2

                        r1 2

        .ADD_TER        t1 1 gnd

        .TER            c1 1

        .ADD_TER        c2 1 vcc

        .TER            r1 1

.END

END.
```

Netzliste (**design.con**) im MARCONI-Format:

```
LAYOUT board;

PARTS

        c1 : cap50;

        c2 : cap75;

        r1 : res;

        t1 : tebc;

MARCONI

        c2 2 t1 3 ; net1 /

        c1 2 t1 2 r1 2 ; net2 /

        t1 1 c1 1 ; gnd /

        c2 1 r1 1 ; vcc /

END.
```

Alle oben aufgeführten Netzlisten können mit folgendemAufruf transferiert werden:

```
>  conconv design laylib ↵
```

Obiger Aufruf bewirkt, dass die ASCII-Netzliste **design.con** eingelesen wird, deren Einträge in der Bauteilliste gegen die Gehäusebibliothek **laylib.ddb** geprüft werden, und anschließend die Netzliste (mit Bauteilliste) unter dem Namen **board** im Jobfile **design.ddb** abgespeichert wird. Wurde der **CONCONV**-Lauf fehlerfrei beendet, dann kann man nun den **Layouteditor** aufrufen, ein Layout mit dem Elementnamen **board** im Jobfile **design.ddb** erzeugen, und anschließend im Menü Bauteile jeweils mit Naechstes Bauteil die in der Bauteilliste eingetragenen Bauteile in das Layout laden. Zu beachten ist hierbei, dass beim Laden der Bauteile dieselbe Bibliothek **laylib.ddb** verwendet wird, wie beim **CONCONV**-Aufruf spezifiziert (evtl. Einstellung im **Layouteditor** über Einstellungen - Bibliotheksname).

Siehe auch

NETCONV

Diagnose

Die durch **CONCONV** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

In den Eingabedaten sind Bauteilnamen, Pinnamen oder Netznamen mit Sonderzeichen (**-**, **+**, **/**, **(**, **=**, usw.) in einfachen oder doppelten Anführungszeichen anzugeben.

7.6 COPYDDB

Name

copyddb - Copy Design Database Utility

Synopsis

```
copyddb srcfile dstfile {-ms|-md|-mr}

        {-a|-as|-al|-ac|-sp|-ss|-sl|-sm|-ll|-lp|-ls|-ld|-cl|-cc|-cp|

         -drc|-llib|-gtab|-fnt|-sct|-lct|-ict|-ulp|-ull|-rule|-recover} [pattern]
```

Beschreibung

Das Programm **COPYDDB** kopiert selektierbare Datenbankeinträge von einem DDB(Design DataBase)-File in ein anderes. Damit lässt sich **COPYDDB** als im Batch-Mode betreibbares Programm zum Mischen von Bibliotheken oder zur Aktualisierung von Job-Design-Files verwenden.

COPYDDB erwartet zwei Dateinamen als Argumente. **`srcfile`** und **`dstfile`** sind die Namen der DDB-Quell- bzw. der DDB-Zieldatei (diese Dateien müssen mit der Extension **`.ddb`** verfügbar sein, wenn die Dateinamen beim Programmaufruf ohne Extension angegeben wurden). **COPYDDB** kopiert Einträge aus der Quelldatei in die Zieldatei, wobei über den Merge-Switch gesteuert werden kann, ob evtl. in der Zieldatei bereits vorhandene Einträge überschrieben werden dürfen oder nicht, bzw. ob lediglich in der Zieldatei bereits vorhandene Elemente ersetzt werden sollen (replace).

COPYDDB erwartet optional ein Schlüsselwort als Argument. Über dieses **`pattern`** werden die Namen der zu kopierenden Elemente spezifiziert. Die Angabe des Schlüsselwortes kann mit Wildcards erfolgen. Ist kein Schlüsselwort angegeben, dann werden alle Elemente der selektierten Klasse kopiert. Die Selektion der zu kopierenden Objektklasse erfolgt über den Class Switch.

Optionen

Merge Switch (erforderlich):

-ms	Merge Source (Quelldatei ist Master)
-md	Merge Destination (Zieldatei ist Master)
-mr	Merge Replace (Quelldatei ist Master)

Class Switch (erforderlich):

-a	Alle Klassen
-as	Alle SCM-Klassen (wie alle **-s?**-Switches)
-al	Alle Layout-Klassen (wie alle **-l?**-Switches)
-ac	Alle Chip/IC Design-Klassen (wie alle **-c?**-Switches)
-sp	SCM Pläne (mit Bauteilliste und logischer Netzliste)
-ss	SCM Symbole (mit Logischer Bibliothek)
-sl	SCM Labels
-sm	SCM Marker
-ll	Layout Pläne (mit physikalischer Netzliste und Routingdaten)
-lp	Layout Bauteile
-ls	Layout Padstacks
-ld	Layout Pads
-cl	Chip/IC Design Layouts (mit physik. Netzliste und Routingdaten)
-cc	Chip/IC Design Zellen
-cp	Chip/IC Design Pins
-drc	Layout Design Rule Check Parameterblöcke
-llib	Logische Bibliothekseinträge
-gtab	Gerber Blendentabellen
-fnt	BAE-Zeichensatzdaten
-sct	SCM Farbtabellen
-lct	Layout Farbtabellen
-ict	Chip/IC Design Farbtabellen
-ulp	User Language-Programme
-ull	User Language-Libraries
-rule	Regelsystemdefinitionen
-recover	Alle Klassen (Restaurierungsmodus für korrupte DDB-Dateien)

Beispiele

Kopieren aller Layoutsymbole deren Namen mit **so1** beginnen (z.B. **so14**, **so16**, ...) aus **newlib.ddb** nach **laylib.ddb** mit **laylib.ddb** als Master-File:

```
>  copyddb newlib laylib -md -lp so1* ↵
```

Kopieren aller SCM-Einträge aus **design.ddb** nach **redesign.ddb** mit **design.ddb** als Master-File:

```
>  copyddb design redesign -ms -as ↵
```

Erneuern der in **design.ddb** enthaltenen Padstack-Definitionen, deren Namen mit **finger** beginnen, entsprechend den in **laylib.ddb** enthaltenen Definitionen:

```
>  copyddb laylib design -ms -ls finger* ↵
```

Kopieren aller **User Language**-Libraries aus **ulcprog.vdb** nach **ullibs.sav** mit **ulcprog.vdb** als Master-File:

```
>  copyddb ulcprog.vdb ullibs.sav -ms -ulp ↵
```

Ersetzen von in **design.ddb** enthaltenen Layoutbauteilen durch entsprechend benannte Layoutbauteile aus **library.ddb**, d.h. Aktualisieren der jobspezifischen Layoutbibliothek in **design.ddb**:

```
>  copyddb library design -mr -lp ↵
```

Dateien

bae.log -- Logfile (im aktuellen Verzeichnis)

Siehe auch

LISTDDB.

Die Funktionalität zum Kopieren von DDB-Dateielementen ist auch in der **User Language**-Systemfunktion **ddbcopyelem** implementiert.

Diagnose

Die durch **COPYDDB** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

COPYDDB ist ein mächtiges Werkzeug zur Manipulation von DDB-Dateiinhalten. Achten Sie ganz besonders darauf, *WAS* Sie mit **COPYDDB** tun. Konflikte könnten auftreten, wenn unterschiedliche SCM- und/oder Layoutpläne gemischt werden, da dadurch auch Bauteil-Listen, Netzlisten, Bauteil-Attribute gemischt werden. Die unsachgerechte Anwendung von **COPYDDB** kann insbesondere zu Problemen im **Packager** bzw. in der **Backannotation** führen. Wir weisen daher mit Nachdruck darauf hin, nach dem Gebrauch von **COPYDDB** den Inhalt der Zieldatei auf Konsistenz zu überprüfen!

7.7 FONTCONV

Name

fontconv - Font Conversion Utility

Synopsis

```
fontconv fontfile libraryfile
```

Beschreibung

Das Programm **FONTCONV** dient dazu, im ASCII-Format erstellte Vektorfontdaten in eine Fontbibliothek zu übertragen.

Das Argument **`fontfile`** gibt den Namen der Fontbeschreibungsdatei an (diese muss mit der Extension **`.fon`** verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben).

Mit dem Argument **`libraryfile`** wird die Bibliothek angegeben, in der die Fontdaten eingetragen werden. Die Bibliotheksdatei erhält die Extension **`.fnt`**, die beim Aufruf nicht mit anzugeben ist. Die Fontbibliothek des **Bartels AutoEngineer** trägt den Namen **`ged.fnt`** und befindet sich im gleichen Verzeichnis wie die ausführbaren Programmdateien.

Format der Eingabedatei

Die Fontbeschreibungsdatei ist nach folgendem Schema aufgebaut:

```
FONT <fontname>;

CHAR <ord>;

        POLY (0, 0), (10, 10), (10, 0) ;

        :

:
END.
```

`<fontname>` gibt den Namen des Fonts in der Bibliothek an. **`<ord>`** gibt die ASCII-Ordnungszahl des definierten Zeichens an (z.B. 65 für **`A`**). Es können Zeichen im Bereich von 0..255 definiert werden. Werden in einer Beschreibungsdatei für die gleiche ASCII-Nummer verschiedene Zeichenbeschreibungen angegeben, so gilt die jeweils letzte. Die Zeichen (CHARacter) werden als Liste von Linienzügen (POLYgonen) beschrieben. Ein Polygon wird durch Aufzählung der Koordinaten seiner Eckpunkte angegeben. Die Koordinaten sind ganzzahlig in einem 32x48 Raster anzugeben, ausgehend von 0,0 in der linken unteren Ecke. Es ergibt sich also ein Wertebereich von 0..31 für X-Koordinaten und 0..47 für Y-Koordinaten. Ein Zeichen kann bis zu 32 Eckpunkte enthalten. Kommentare beliebiger Länge können zwischen den Zeichenketten **`/*`** und **`*/`** an beliebiger Stelle eingefügt werden.

Beispiele

Inhalt der Datei **test.fon** mit Definition für **!** und **"** (je 4 Eckpunkte und 2 Polygone):

```
/* Name des Fonts in der Bibliothek */

FONT test;

/* ASCII-Code 33 fuer '!' */

CHAR 33;

        /* Kurzer unterer Strich */

        POLY (16,5),(16,9);

        /* Langer oberer Strich */

        POLY (16,13),(16,45);

/* ASCII-Code 34 fuer '"' */

CHAR 34;

        /* Linker Strich */

        POLY (12,40),(4,32);

        /* Rechter Strich */

        POLY (16,32),(24,40);
END.
```

Eintragen in die Fontbibliothek (**ged.fnt**) mit:

```
>  fontconv test ged ↵
```

Dateien

ged.fnt -- BAE-Fontbibliothek (im Programmverzeichnis)

Siehe auch

FONTEXTR

Diagnose

Die durch **FONTCONV** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

Ein eventuell schon in der Bibliothek enthaltener Font gleichen Namens wird kommentarlos ersetzt.

7.8 FONTEXTR

Name

fontextr - Font Extraction Utility

Synopsis

```
fontextr fontname libraryfile
```

Beschreibung

Das Programm **FONTEXTR** dient dazu, Vektorfontdaten aus dem internen Bibliotheksformat in ein les- und editierbares ASCII-Format umzuwandeln.

Das Argument **`fontname`** gibt den Namen des Fonts in der Bibliothek an. Die ausgelesenen Fontdaten werden in einer ASCII-Textdatei mit dem Namen **`<fontname>.fon`** abgelegt.

Mit dem Argument **`libraryfile`** wird die Bibliothek angegeben, aus der die Fontdaten ausgelesen werden. Die Bibliotheksdatei hat die Extension **`.fnt`**, die beim Aufruf nicht mit anzugeben ist. Die Fontbibliothek des Bartels AutoEngineers trägt den Namen **`ged.fnt`** und befindet sich im gleichen Verzeichnis wie die ausführbaren Programmdateien.

Format der Ausgabedatei

Die Fontbeschreibungsdatei ist nach folgendem Schema aufgebaut:

```
FONT <fontname>;

CHAR <ord>;    /* 'ASCII-Character' */

        POLY (0, 0), (10, 10), (10, 0) ;

        :

:
END.
```

`<fontname>` gibt den Namen des Fonts in der Bibliothek an. **`<ord>`** gibt die ASCII-Ordnungszahl des definierten Zeichens an, z.B. 65 für **`A`**. Die Zeichen (CHARacter) werden als Liste von Linienzügen (POLYgonen) beschrieben. Ein Polygon wird durch Aufzählung der Koordinaten seiner Eckpunkte angegeben. Am Ende der Kopfzeile jeder Zeichendefinition steht in Kommentarform die ASCII-Repräsentation des Zeichens.

Beispiele

Auslesen des Fonts **`standard`** aus der Fontbibliothek **`ged.fnt`** (und Ablegen der Zeichensatzdaten in der ASCII-Datei **`standard.fon`**):

```
>  fontextr standard ged ↵
```

Dateien

`ged.fnt` -- BAE-Fontbibliothek (im Programmverzeichnis)

Siehe auch

FONTCONV

Diagnose

Die durch **FONTEXTR** erzeugten Fehlermeldungen sind selbsterklärend.

7.9 INSTALL

Name

install - Bartels AutoEngineer Installation Utility

Synopsis

```
install

install [-c] srcfile[pattern] dstfile[directory\*]
```

Beschreibung

Mit dem Programm **INSTALL** kann die PC-Software des **Bartels AutoEngineer** ganz oder teilweise vom Installationsmedium der BAE-Software auf die PC-Festplatte installiert werden. Darüber hinaus lassen sich mit **INSTALL** auch einzelne oder mehrere Dateien entweder komprimieren oder dekomprimieren.

Die Installation der BAE-PC-Software kann nur mit diesem Utilityprogramm durchgeführt werden, da die Dateien auf den Installationsdisketten des **Bartels AutoEngineer** in einem komprimierten Format ausgeliefert werden.

BAE-Softwareinstallation

Die Installation der BAE-PC-Software kann mit dem Aufruf

```
install
```

gestartet werden. Voraussetzung hierfür ist, dass das aktuelle Verzeichnis den Inhalt des BAE-Install-Kits enthält (dies ist z.B. dann der Fall, wenn die Disk 1 des BAE-Install-Kits in einem Diskettenlaufwerk oder die BAE-CD-ROM im CD-ROM-Laufwerk eingelegt ist, und wenn das Arbeitsverzeichnis mit dem entsprechenden Laufwerk übereinstimmt). Nach Aufruf von **INSTALL** durchläuft der Benutzer eine Abfragesequenz, in der u.a. der Installationsmodus sowie die Zielverzeichnisse zur Installation der BAE-Softwarekomponenten abgefragt werden. Der Installationsmodus gibt an, welche Dateien kopiert werden sollen. Bei einer Neu-Installation werden alle Dateien kopiert. Bei einer Update-Installation werden auf **`.dat`**, **`.def`** und **`.fnt`** endende Systemdateien mit benutzerspezifischen Setupdaten, Farb- oder Blendentabellen, Zeichensätzen, usw. nicht überschrieben. Bei der Abfrage der Verzeichnisse werden Defaultnamen angezeigt. Diese können durch einfaches Drücken der Eingabetaste ↵ übernommen werden. Die Pfadnamen der Zielverzeichnisse können editiert werden. Wahlweise können Zielverzeichnisnamen auch gelöscht werden, um die Installation der entsprechenden Softwarekomponenten zu unterdrücken. Ist eines der Zielverzeichnisse nicht vorhanden, dann wird es während der Installation nach Bestätigung durch den Benutzer automatisch angelegt.

Komprimieren/Dekomprimieren von Dateien

Selektierbare Dateien lassen sich mit der Aufrufart

```
install [-c] srcfile[pattern] dstfile[directory\*]
```

dekomprimieren (ohne Option **`-c`**) bzw. komprimieren (mit Option **`-c`**), wobei die Spezifikation von Wildcards optional zulässig ist (**`pattern`**). Ist das Ziel ein Verzeichnis für mehrere über Wildcard spezifizierte Dateien, so muss **`\*`** an den Verzeichnisnamen angehängt werden.

Beispiele

Dekomprimieren aller **.ddb**-Dateien von Floppy/Laufwerk A und Kopieren in ein Verzeichnis auf Festplatte:

```
> install a:\*.ddb c:\baelib\* ↵
```

Dekomprimieren der Datei **ged.fnt** von Floppy/Laufwerk B und Kopieren in ein Verzeichnis auf Festplatte:

```
> install b:\ged.fnt c:\bae ↵
```

Komprimieren der Datei **design.ddb** und Ablegen der komprimierten Datei unter **design.cmp**:

```
> install -c design.ddb design.cmp ↵
```

Diagnose

Die durch **INSTALL** erzeugten Fehlermeldungen sind selbsterklärend.

7.10 LISTDDB

Name

listddb - List Design Database Utility

Synopsis

```
listddb ddbfile listfile
```

Beschreibung

Mit dem Programm **LISTDDB** kann der Inhalt von DDB(Design DataBase)-Files ausgelistet werden.

Das Argument **`ddbfile`** gibt den Namen der DDB-Datei an, deren Inhalt ausgelistet werden soll (die DDB-Datei muss mit der Extension **`.ddb`** verfügbar sein, wenn der Dateiname beim Programmaufruf ohne Extension angegeben wurde).

Das Argument **`listfile`** gibt den Namen der ASCII-Datei an, auf die der Inhalt der DDB-Datei ausgelistet werden soll.

Beispiele

Auflisten des Inhalts von **`laylib.ddb`** auf die Datei **`laylib.lst`**:

```
> listddb laylib laylib.lst ↵
```

Siehe auch

COPYDDB

Diagnose

Die durch **LISTDDB** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

Das **`listfile`** Argument gibt den Namen der durch **LISTDDB** zu erzeugenden Ausgabedatei an. **LISTDDB** führt *keine* Prüfung auf Existenz dieser Datei durch. Geben Sie daher in keinem Fall den Namen einer noch benötigten existierenden Datei an, da diese sonst durch **LISTDDB** kommentarlos überschrieben wird!

7.11 LOGLIB

Name

loglib - Logical Library Maintenance

Synopsis

```
loglib loglibfile libraryfile
```

Beschreibung

Das Programm **LOGLIB** dient dazu, Informationen über die Zuordnung der Stromlaufsymbole zu den Layoutsymbolen, die Zuordnung von logischen zu physikalischen Anschlüssen, Pin- und Gattertausch, feste Versorgungsanschlüsse, usw. in ein DDB(Design DataBase)-File einzuspielen. Dies ist notwendig, um die Transformation von Schaltplänen in das Layout, d.h. die Umsetzung logischer Netzlisten in physikalische (mit Hilfe des **Packager**s oder umgekehrt über Backannotation) zu ermöglichen.

LOGLIB erwartet als erstes Argument den Namen **`loglibfile`** einer ASCII-Datei (Loglibdatei), in der die logische Bibliotheksinformation abgelegt ist (diese Datei muss mit der Extension **`.def`** verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben).

LOGLIB erwartet als zweites Argument den Namen **`libraryfile`** der DDB-Datei, in das die logische Bibliotheksinformation einzuspielen ist (diese Datei erhält die Extension **`.ddb`** beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben).

Gebrauch

Das Programm **LOGLIB** muss immer zur Anwendung kommen, wenn ein neues (oder geändertes) Stromlaufsymbol definiert wurde bzw. wenn sich eine neue (oder geänderte) Zuweisung von Stromlauf- zu Layoutsymbol ergibt. Im Regelfall wird zunächst das Stromlaufsymbol in der Stromlaufbibliothek erstellt (oder editiert). Anschließend wird (falls nicht schon existent) in der Layoutbibliothek das Layoutsymbol definiert, in welches das Stromlaufsymbol gepackt werden soll. Schließlich sind die entsprechenden Einträge in einer Loglibdatei zu erstellen und die darin eingetragenen Definitionen mit dem Programm **LOGLIB** in die Layoutbibliothek einzuspielen.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Loglibdatei beginnt mit dem Schlüsselwort **loglib** und endet mit **end.**. Kommentare können zwischen **/*** und ***/** eingefügt werden.

Kommando part

Über das **part**-Kommando erfolgt die Zuweisung des Stromlaufsymbols an das Layoutsymbol. Die Syntax für dieses Kommando lautet:

```
part <llname> : <plname>
```

Der **Packager** wird bei der Umsetzung des Stromlaufs in das Layout entsprechend des eingetragenen **part**-Kommandos das durch **<llname>** (Logical Library Name) gegebene Stromlaufsymbol in die durch **<plname>** (Physical Library Name) gegebene Gehäuseform packen. Steht vor **<plname>** das Schlüsselwort **default**, so besteht die Möglichkeit, bei der Stromlauf-Erstellung über einen entsprechenden Werteintrag für das am Stromlaufsymbol **<llname>** definierte Attribut **$plname** dem **Packager** eine andere Gehäusezuweisung vorzugeben:

```
part <llname> : default <plname>
```

Wird für **<plname>** eine (durch Komma getrennte) Liste von Gehäusen angegeben, dann besteht während der Bauteilplatzierung im Layout die Möglichkeit der Zuweisung der entsprechenden Alternativbauform(en) an das betreffende Bauteil. Die Syntax hierfür lautet:

```
part <llname> : [default] <plname>[,<plname>,...,<plname>]
```

Der erste **<plname>**-Eintrag entspricht dabei dem Default-Gehäuse für das entsprechende Bauteil. Die nachfolgenden Einträge spezifizieren die Alternativbauformen, die über die Funktion Bauform aendern während der Platzierung des entsprechenden Bauteils gewählt werden können. Die Auswahlreihenfolge im Menü zur Bauformauswahl entspricht hierbei der Reihenfolge der **<plname>**-Einträge im **part**-Kommando. Im Layout durchgeführte Zuweisungen alternativer Gehäusebauformen müssen mit Backannotation in den Stromlauf zurückgemeldet werden. In jedem Fall haben jedoch im Stromlauf über das Attribut **$plname** definierte Gehäusezuweisungen höhere Priorität.

Mit Hilfe des Schlüsselwortes **class** kann das Bauteil wahlweise einer Bauteilklasse zugewiesen werden:

```
part <llname> : class <partclassname> [default] <plname>
```

Bauteilklassen werden vom **Packager** ausgewertet, um die Zulässigkeit der Zuweisung zu alternativen Bauteildefinitionen über das Attribut **$rlname** (Requested Logical Library Name) zu prüfen.

Soll eine einmal über **LOGLIB** vorgenommene Zuweisung vollständig aus der Bibliothek gelöscht werden, so kann dies mit dem Kommando

```
delete part <llname> ;
```

geschehen. Die Loglibdatei bietet außerdem die Möglichkeit virtuelle Symbole zu definieren, also Stromlaufsymbole, die keinem Layoutsymbol zuzuordnen sind. In diesem Fall ist im **part**-Kommando anstelle des **<plname>** das Schlüsselwort **virtual** einzutragen:

```
part <llname> : virtual ;
```

Das Kommando **part** ermöglicht über das Schlüsselwort **logical** die Definition rein logischer Bauteile:

```
part <llname> : logical ...
```

Die Definition logischer Bauteile ohne Gehäusezuweisung ermöglicht die Generierung logischer Netzlisten (z.B. im Format EDIF für PLD- bzw. LCA-Design).

Die Zuweisung unterschiedlicher Stromlaufsymbole in ein einziges Gehäuse (z.B. Spule und Kontakt beim Relais) geschieht durch Verwendung der Schlüsselwörter **mainpart** und **subpart** in den entsprechenden **part**-Kommandos:

```
part <mainllname> : mainpart <plname>

:
part <subllname> : subpart <mainllname>
```

Bei der **subpart**-Definition ist zu beachten, dass hier anstelle eines Verweises auf einen **<plname>** ein Verweis auf den **<llname>** des **mainpart** (**<mainllname>**) einzutragen ist.

Für den Fall, dass eine 1:1-Zuordnung zwischen den logischen Pins und den physikalischen Anschlüssen vorliegt (d.h. das Stromlaufsymbol ist genau einem Layoutsymbol zugeordnet, und die Pinbezeichnungen entsprechen sich), dann genügt es, das **part**-Kommando mit einem Strichpunkt abzuschließen:

```
part <llname> : <plname> ;
```

Liegt der Sonderfall der 1:1-Zuordnung nicht vor, oder sind Pin- und Gattertausch, Versorgungsanschlüsse, feste Attribute oder ähnliches zu definieren, dann geschieht dies mit den entsprechenden, nachfolgend beschriebenen Kommandos unmittelbar nach dem **part**-Kommando in geschweiften Klammern:

```
part <llname> : <plname> { <commands> }
```

Kommando net

Über das Kommando **net** können Netze definiert werden, die im Schaltbild immer wieder auftreten und deshalb nicht einzeln verdrahtet werden müssen (z.B. feste Versorgungsanschlüsse). Die Syntax hierfür lautet:

```
net "<netname>" : ( <pinlist> ) ;
```

Durch Voranstellen des Dollarzeichens (**$**) kann mit dem Kommando **net** ein Netznamensattribut anstelle des Netznamens definiert werden:

```
net "$<netname>" : ( <pinlist> ) ;
```

Ist auf dem zugehöhrigen SCM-Symbol ein entsprechendes Attribut definiert, dann kann im Stromlauf eine variable, bauteilspezifische Versorgungsspannungszuweisung durch einen entsprechenden Netznamenseintrag (z.B. **vcc**, **vss**, **0v**, etc.) für das Netznamensattribut vorgenommen werden.

Eine Sonderform des **net**-Kommandos ist das **internal**-Kommando, in dem anstelle einer Netznamensdefinition lediglich das Schlüsselwort **internal** eingetragen ist:

```
net internal : ( <pinlist> ) ;
```

Das **internal**-Kommando sorgt dafür, dass die in der zugehörigen **<pinlist>** definierten Anschlüsse durch den **Packager** automatisch miteinander verbunden werden. **<pinlist>** enthält (getrennt durch Komma) die Liste der Pins. Die **pin**-Einträge definieren die physikalischen Pinbezeichnungen. Enthält ein Pinname Sonderzeichen, dann ist er in Anführungszeichen anzugeben.

Kommando bus

Mit Hilfe des **bus**-Kommandos können bereits direkt am Bauteil Busse definiert werden. Die Syntax für dieses Kommando lautet:

```
bus ( <buspinlist> ) ;
```

<buspinlist> definiert hierbei die Liste der am Bauteil definierten logischen Pins, über die jeweils ganze Busse angeschlossen werden können. Die Definition der jeweiligen Bussignale kann dann z.B. mit dem **xlat**-Kommando (siehe unten) erfolgen. Dabei ergeben sich die entsprechenden logischen Pinbezeichnungen jeweils aus dem Bus(pin)namen und dem Bussignalnamen in folgendem Format:

```
"<buspin>.<bussignal>"
```

Kommando pin

Das Kommando **pin** dient der Auflistung der am Stromlaufsymbol definierten logischen Anschlüsse. Die Syntax für dieses Kommando lautet:

```
pin ( <pinlist> ) ;
```

<pinlist> enthält hierbei die Liste der logischen Pinbezeichnungen am Stromlaufsymbol.

Bei der Angabe von Pinnamen lassen sich optional auch ganze Bereiche in der Form

```
<startpin>-<endpin>[:<schrittweite<]
```

angeben. So kann z.B. die Pinliste **pin(a1,a2,a3,a4)** mit **pin(a1-a4);** oder die Pinliste **pin(c2,c4,c6,c8,c10)** mit **pin(c2-c10:2)** spezifiziert werden. Es sind auch mehrere Bereiche wie z.B. **pin(a1-a32,b1-b2,c1-c32);** in einem Kommando erlaubt. Die Bereichsangabe dient nur der Verkürzung der Eingabe. Beim Abspeichern und der Anzeige mit Symbollogik zeigen werden weiterhin die vollständigen Pinnamenslisten verwendet.

Das Kommando

```
pin none ;
```

unterdrückt die automatische 1:1-Zuordnung von Symbol- zu Layoutbauteilpins bei fehlendem **pin**-Kommando und gestattet somit z.B. die Erstellung universell verwendbarer **mainpart**-Symbole ohne Pins.

Kommando xlat

Über das Kommando **xlat** erfolgt die Zuordnung der (über Kommando **pin** definierten) logischen Pins zu den physikalischen Anschlüssen am Gehäuse, d.h. die Definition der Gatterzuordnung. Die Syntax für das **xlat**-Kommando lautet:

```
xlat ( <lplist> ) to ( <pplist> )
  or ( <pplist> ) or ... or ( <pplist> ) ;
```

<lplist> enthält die Liste der am Stromlaufsymbol definierten logischen Pinbezeichnungen. Die **<pplist>**-Einträge definieren (entsprechend der in **<lplist>** vorgegebenen Reihenfolge) die Zuweisungen der logischen zu den physikalischen Pinbezeichnungen.

Für jedes **xlat**-Kommando mit Alternativen (d.h. mit **or**-Optionen zur Definition von Gattern) wird automatisch eine **swap**-Definition für bauteilübergreifenden Gattertausch generiert, sofern nicht explizit ein **swap internal**-Kommando (siehe unten) mit angegeben wurde.

Kommando swap

Das Kommando **swap** erlaubt die Definition von Pin-, Gatter-, oder Pingruppen-Vertauschbarkeit (vgl. hierzu Funktion Bauteile - Pin/Gate Swap im **Layouteditor**). Die Syntax für das **swap**-Kommando lautet:

```
swap <swapdefinition> ;
```

Die **<swapdefinition>** enthält gestaffelt in bis zu vier Klammerebenen die Vertauschbarkeits-Definitionen, wobei über Pinlisten die physikalischen Anschlüsse der jeweils vertauschbaren Swap-Ebene angegeben sind. Die Funktion der jeweiligen Klammerebene ergibt sich aus folgendem Schema:

(	**Bauteil-Tausch**	**)**
([	**Pingruppentausch**	**])**
([(	**Gattertausch**	**)])**
([((	**Pintausch**	**))])**

Um die Möglichkeit des bauteilübergreifenden Gatter-Tausches zu unterdrücken, kann im **swap**-Kommando hinter dem Schlüsselwort **swap** das Schlüsselwort **internal** eingefügt werden.

Kommando newattr

Mit Hilfe des Kommandos **newattr** können über die Loglibdatei Attribute mit Werteinträgen in die Bibliothek übernommen werden. Die Syntax hierfür lautet:

```
newattr "$<attname>" = "<attvalue>" ;
```

<attname> definiert dabei den Attributnamen, und **<attvalue>** enthält den zugehörigen Attributwert. Derartige Attributwertzuweisungen können im Layout durch die Definition des Textes **<attname>** (z.B. auf einer Dokumentarlage) visualisiert werden. Auch kann der entsprechende Werteintrag mit Hilfe von **User Language**-Programmen bzw. mit dem Utilityprogramm **USERLIST** weiter in Richtung Postprozess, PPS, usw. ausgewertet werden (typische Beispiele: **$sachnummer** für Stücklisten, **$delay** für Simulation, **$bauteilhoehe** für Bestückautomat, **$preis**, **$lieferant**, **$toleranz**, usw.).

Das Kommando **newattr** ermöglicht auch die Definition bzw. Zuweisung pinspezifischer Attribute bzw. Attributwerte. Die Syntax hierfür lautet:

```
newattr "$<attname>" = "<attvalue>" to ( <pplist> ) ;
```

In der Pinliste sind die physikalischen Pinbezeichnungen einzutragen. Damit können z.B. Pintypdefinitionen eingetragen werden wie sie für Electronic Rule Check (ERC) oder zur Generierung von Netzlisten für Simulatoren wie PSpice benötigt werden.

Der **Packager** erlaubt über das Pinattribut **`$pintype`** eine Plausibilitätsprüfung für die im Schaltplan vorgenommenen Verbindungen zwischen Pins verschiedener Typen. Die Pinattribute werden zweckmässigerweise in den logischen Definitionen der Symbole fest für die einzelnen Layoutbauteilpins vergeben. Unterstützt werden die folgenden Pintypen:

`$pintype`	**Pintyp**
`in`	Eingabe-Pin
`out`	Ausgabe-Pin
`bidi`	Bidirektionaler Anschluss
`anl`	Analoger Anschluss
`sup`	Stromversorgungsanschluss

Der ERC überprüft für Netze mit mindestens einem Eingang, ob an diesem Netz ein normaler Ausgang, ein bidirektionaler Anschluss oder ein Versorgungsspannungspin vorhanden ist und gibt ggf. die Warnmeldung **`Netz 'Netzname' hat nur Eingaenge!`** aus. Außerdem überprüft der ERC, ob an einem normalen Ausgang ein anderer Ausgang, ein bidirektionaler Anschluss oder ein Versorgungsspannungspin angeschlossen ist und gibt ggf. die Warnmeldung **`Treiber-Kollision auf Netz 'Netzname'!`** aus.

Das **`newattr`**-Kommandos unterstützt auch die Vergabe variantenspezifischer Attribute. Dazu ist die Variantennummer dem Attributnamen nach dem Anführungszeichen mit Komma getrennt anzuhängen. Damit lassen sich bei fixer Verwendung von Varianten für alle Projekte, wie z.B. **`110 Volt`** und **`230 Volt`** oder **`deutsch`** und **`english`** den einzelnen Varianten unterschiedliche feste Attributwerte zuordnen. Ohne Variantennummernangabe wird der Attributwert der Basisvariante zugeordnet.

Über das Kommando **`newattr`** kann durch Zuweisung der speziellen Werte **`?id?`**, **`?symid?extension`** und **`?partid?extension`** eine automatische Generierung von ID-Attributwerten durch den **Packager** angefordert werden. **`?id?`** erzeugt dabei fortlaufende ID-Werte nach dem Schema **`id1`**, **`id2`** usw., bei **`?symid?extension`** und **`?partid?extension`** wird die gegebene Extension mit Unterstrich an den Schaltplansymbolnamen bzw. den Layoutbauteilnamen des gerade bearbeiteten Symbols angehängt. So ergibt z.B. **`?partid?diffpair1`** ID-Werte nach dem Schema **`ic1_diffpair1`**, **`ic2_diffpair1`** usw. Die automatische ID-Generierung ist nützlich, wenn **`newattr`** auf mehrere Pins verweist, da so automatisch eine Beziehung zwischen Pins hergestellt werden kann, wie sie z.B. für die Markierung von Differential Pairs benötigt wird.

Bei Vorgabe des speziellen Wertes **`!unique!`** gibt das **`newattr`**-Kommando keine Attributzuweisung aus der logischen Definition heraus an, sondern steuert die Gatterzuordnung durch den **Packager** so, dass nur Symbole mit gleichen Werten für die so markierten Attribute zusammen in ein Layoutbauteil gepackt werden. Dabei werden die **`swap`**-Kommandos automatisch so berücksichtigt, dass Gatter bauteilübergreifend nur zwischen Bauteilen mit gleichen Werten für die so markierten Attribute getauscht werden können. Die **`!unique!`**-Einstellung kann somit anstelle der Packungssteuerung über **`$rpname`**-Attribute bei Symboltypen mit unterschiedlichen Attributwerten innerhalb eines Projekts verwendet werden. Dies ist z.B. bei Widerstandsarrays nützlich:

```
part ra_so16r : so16r {

    newattr "$val" = "!unique!";

    pin (1-16);

    swap internal (

        (( 1,16)),(( 2,15)),(( 3,14)),(( 4,13)),

        (( 5,12)),(( 6,11)),(( 7,10)),(( 8, 9))

        );
    }
```

Das folgende Beispiel zeigt die Verwendung des **!unique!**-Werts bei der Definition eines Operationsverstärkers mit Versorgungsspannunszuweisung über Attributwerte:

```
part op_lm324 : dil14 {

    pin (/i,i,o);

    net "$vplus" : (4);

    net "$vminus" : (11);

    newattr "$vplus" = "!unique!";

    newattr "$vminus" = "!unique!";

    xlat (/i, i, o)

      to ( 2, 3, 1)

      or ( 6, 5, 7)

      or ( 9,10, 8)

      or (13,12,14);

    swap internal ((2,3,1),(6,5,7),(9,10,8),(13,12,14));
    }
```

Kommando netattr

Über das Kommando **netattr** kann bereits im Stromlauf die Möglichkeit der Spezifikation von Router-Steuerparametern angeboten werden. Die Syntax für das Kommando **netattr** lautet:

```
netattr <netatt> "$<attname>" : ( <pinlist> ) ;
```

Für **<netatt>** können beliebige Attributnamen eingesetzt werden, wobei jedoch folgende Schlüsselworte spezielle Attribute zur **Autorouter**-Steuerung definieren, d.h. vom System in spezieller Weise ausgewertet werden:

routwidth	netzbezogene Leiterbahnbreite (in mm)
powwidth	pinbezogene Anschlussbreite für die über Kommando **net** definierten Signale (in mm)
mindist	netzbezogener Mindestabstand zu anderen Potentialen (in mm)
priority	netzbezogene Routing-Priorität (Integer; je größer, desto höher die Priorität)

Die mit dem **AutoEngineer** gelieferte Bibliothek enthält in **route.ddb** virtuelle Stromlaufsymbole, über die **netattr**-Einträge im Stromlauf möglich sind. Die entsprechende Loglibdatei (**route.def**) hat folgenden Inhalt:

```
loglib

part att_rw : virtual

{

    pin (x);

    netattr routwidth "$val" : (x);

}

part att_pw : virtual

{

    pin (x);

    netattr powwidth "$val" : (x);

}

part att_md : virtual

{

    pin (x);

    netattr mindist "$val" : (x);

}

part att_pr : virtual

{

    pin (x);

    netattr priority "$val" : (x);

}

end.
```

Wie aus obiger Loglibdatei zu ersehen ist, enthalten die virtuellen Symbole jeweils einen Pin (benannt **x**) und ein Attribut. Die Zuweisung des jeweiligen Attributes an ein bestimmtes Signal erfolgt im Stromlauf-Editor durch Anschluss des virtuellen Symbols (über Pin **x**) an das betreffende Signal sowie Eintrag des entsprechenden Attributwertes.

Durch die Möglichkeit der Definition beliebiger anwendungsspezifischer Netzattribute lassen sich zusätzliche Signalnetzinformationen auf gesonderte Weise verarbeiten bzw. auswerten. Solche Netzattribute lassen sich z.B. zur Kontrolle des Layoutprozesses (maximal/minimal zulässige Leiterbahnlänge, Parallelführung von Leiterbahnen, Lagenzuordnung, usw.) verwenden oder können zur Steuerung nachgeschalteter Simulationsprozesse bzw. Laufzeitanalysen oder zur Prüfung von ECL/EMV-Regeln herangezogen werden (tatsächliche Leiterbahnlängen, Parallelverlauf von Leiterbahnen, usw.). Die Bereitstellung entsprechender Tools zur Auswertung anwendungsspezifischer Netzattribute kann mit Hilfe entsprechender **User Language**-Programme erfolgen.

Über das Kommando **netattr** kann durch Zuweisung der speziellen Werte **?id?**, **?symid?extension** und **?partid?extension** eine automatische Generierung von ID-Attributwerten durch den **Packager** angefordert werden. **?id?** erzeugt dabei fortlaufende ID-Werte nach dem Schema **id1**, **id2** usw., bei **?symid?extension** und **?partid?extension** wird die gegebene Extension mit Unterstrich an den Schaltplansymbolnamen bzw. den Layoutbauteilnamen des gerade bearbeiteten Symbols angehängt. So ergibt z.B. **?partid?diffpair1** ID-Werte nach dem Schema **ic1_diffpair1**, **ic2_diffpair1** usw. Die automatische ID-Generierung ist nützlich, wenn **netattr** auf mehrere Netze verweist, da so automatisch eine Beziehung zwischen Netzen hergestellt werden kann, wie sie z.B. für die Markierung von Differential Pairs benötigt wird.

Kommando call

Über das **call**-Kommando erfolgt die Zuweisung von Blockstromlaufplänen an Stromlaufsymbole für hierarchisches Schaltplandesign. Die Syntax hierfür lautet:

```
call <blockname> ;
```

Die Definition des Bauteiles erfolgt dabei **virtual**. Die Pins des Stromlaufsymbols werden den gleichnamigen Modulports in den Blockstromlaufplänen zugewiesen.

Kommando architecture

Über das Kommando **architecture** lassen sich Stromlaufsymbole (vom Typ **virtual**) aus intern beliebig verschalteten anderen Stromlaufsymbolen aufbauen, die zudem mehreren Layoutbauteilen zugeordnet sein können. Die Syntax hierfür lautet:

```
architecture { <partlist> }
```

<partlist> enthält eine Auflistung der verwendeten Symbole jeweils mit in Klammern gesetzter kommaseparierter Auflistung der Symbolpins in der Form:

```
<pinname:connection>
```

Hierbei kann **<connection>** der Name eines Pins des **<architecture>**-Symbols sein. In der Form **<net netname>** kann der Anschluss an ein global benanntes Netz erfolgen. Mit **<net & intnetname>** oder **<& intnetname>** kann ein nur lokal für das **<architecture>**-Symbol gültiges Netz referenziert werden.

Beispiele

Loglibdatei **`example.def`** mit der Definition dreier Bauteile:

```
loglib

/* Example Loglib File */

part 74ls00 : dil14, so14

{

    newattr "$partnumber" = "A-NAND-X11B82";

    newattr "$pintype" = "in" to (1,2,4,5,9,10,12,13);

    newattr "$pintype" = "out" to (3,6,8,11);

    newattr "$pintype" = "sup" to (7,14);

    pin (a,b,y);

    net "vcc" : (14);

    net "$groundnetname" : (7);

    xlat ( a, b, y)

      to ( 1, 2, 3)

      or ( 4, 5, 6)

      or (13,12,11)

      or (10, 9, 8);

    swap (((1,2),3),((4,5),6),((13,12),11),((10,9),8));

}

part tx27 : default sot23;

part tr_bc547 : class "npn-transistor" default to92

{

    pin (e,b,c);

    xlat (e,b,c)

      to (1,2,3);

}

end.
```

In obigem Beispiel ist das Gatter **74ls00** definiert, das bis zu viermal in das Gehäuse **dil14** gepackt werden kann, wobei die logischen Pins (a,b,y) wahlweise den physikalischen Anschlüssen (1,2,3), (4,5,6), (13,12,11) oder (10,9,8) zugewiesen werden können. Dieses Gehäuse kann im Layout gegen die Alternativbauform **so14** getauscht werden. Außerdem ist für das Bauteil **74ls00** das Attribut mit Namen **partnumber** und Wert **A-NAND-X11B82** definiert, und jedem physikalischen Pin ist ein Attribut zur Spezifikation des Pintyps zugewiesen. Der Anschluss 14 des Gehäuses **dil14** ist fest mit dem Signal **vcc** verbunden, während der Anschluss 7 auf Stromlaufebene durch Zuweisung eines Netznamenseintrags (z.B. **vss**) an das Attribut **$groundnetname** des Symbols **74ls00** mit dem gewünschten Signalnetz verbunden werden kann. Über das **swap**-Kommando sind für das Gehäuse **dil14** mit den vier Gattern **74ls00** folgende Tauschmöglichkeiten definiert:

```
Pintausch: (1,2) bzw. (4,5) bzw. (13,12) bzw. (10,9)

Gattertausch: (1,2,3) mit (4,5,6) bzw. (13,12,11) bzw. (10,9,8)
Bauteil-Tausch: (1,2,3,4,5,6,13,12,11,10,9,8)
```

Neben dem Bauteil **74ls00** sind in obigem Beispiel auch noch das Bauteil **tx27** (per Default direkt dem Gehäuse **sot23** zugeordnet) sowie der NPN-Transistor **tr_bc547** (mit Zuweisung an Bauteilklasse **npn-transistor**) definiert.

Obige Loglibdatei **example.def** lässt sich mit folgendem **LOGLIB**-Aufruf in die DDB-Datei **mylib.ddb** einspielen:

```
>  loglib example mylib ↵
```

Folgendes Beispiel zeigt die Zuweisung verschiedener Stromlaufsymbole an ein einziges Gehäuse (in diesem Fall geschieht dies für ein Relais-Bauteil mit zwei Kontakten und einer Spule):

```
loglib

/* Relais-Bauteil */

part rel2 : mainpart dilrel

{

    xlat (a,b) to (1,7) or (8,14);

    swap ((1,7),(8,14));

}

part rel2sub : subpart rel2

{

    xlat (p,m) to (2,6);

}

end.
```

Folgendes Beispiel zeigt die Definition der beiden Busse **`b1`** und **`b2`** mit den Bussignalen **`0`**, **`1`**, **`2`** und **`3`** am Bauteil **`buspart`** (das entsprechende Stromlaufsymbol **`buspart`** besitzt in diesem Fall die beiden Pins **`b1`** und **`b2`**, von wo aus alle definierten Bussignale über entsprechende Busverbindungen abgegriffen werden können):

```
loglib

/* Bus-Bauteil */

part buspart : sot8

{

    bus (b1,b2);

    xlat (b1.0,b1.1,b1.2,b1.3) to (1,2,3,4);

    xlat (b2.0,b2.1,b2.2,b2.3) to (5,6,7,8);

}

end.
```

Folgendes Beispiel zeigt die Definition eines Symbols für das hierarchische Schaltplandesign. Dem Stromlaufsymbol **`dff`** werden dabei die zugehörigen Blockstromlaufpläne **`dff`** zugewiesen. Die Pins **`s`**, **`r`**, **`q`** und **`/q`** müssen auf den entsprechenden Blockstromlaufplänen als Modulports definiert sein.

```
loglib

/* Hierarchisch aufgebautes Flip-Flop */

part dff : virtual

{

    pins (s,r,q,/q);

    call dff;

}

end.
```

Folgendes Beispiel zeigt eine synthetische Symboldefinition über das Kommando `architecture`. Das Stromlaufsymbol `delay` wird dabei intern aus einer Serienschaltung von vier `74ls04`-Invertern aufgebaut, die über lokale interne Netze miteinander verbunden sind.

```
loglib

/* Synthetisch generierte Inverter-Serienschaltung */

part delay : virtual

{

    pin (in,out);

    architecture

    {

        part "74ls04" (a:in,y:&connect1);

        part "74ls04" (a:&connect1,y:&connect2);

        part "74ls04" (a:&connect2,y:&connect3);

        part "74ls04" (a:&connect3,y:out);

    }

}

end.
```

Dateien

Das bei der Software-Installation des **Bartels AutoEngineer** generierte Bibliotheks-Verzeichnis enthält neben der Layoutbibliothek `laylib.ddb` für alle mitgelieferten Stromlauf-Bibliotheksdateien die entsprechenden Loglibdateien (`*.def`), die alle bereits mit dem Programm **LOGLIB** in die Layoutbibliothek eingespielt sind.

Siehe auch

Packager.

Die Funktionalität zur Kompilierung logischer Bibliotheksdefinitionen ist auch in der **User Language**-Systemfunktion **con_compileloglib** implementiert.

Diagnose

Die durch **LOGLIB** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

In den Eingabedaten sind Bauteilnamen, Pinnamen oder Netznamen mit Sonderzeichen (**`-`**, **`+`**, **`/`**, **`(`**, **`=`**, usw.) in einfachen oder doppelten Anführungszeichen anzugeben.

Wird im Stromlauf oder im Layout während der Bearbeitung ein Symbol aus einer Bibliotheksdatei in eine Projektdatei geladen, dann wird datentechnisch eine komplette Kopie des Symbols in der Projektdatei abgelegt. Dies führt dazu, dass im Laufe der Bearbeitung *in jeder Projektdatei* automatisch eine *projektspezifische Bibliothek* angelegt wird. Wird über eine Loglibdatei eine Veränderung vorgenommen, von der Layoutsymbole betroffen sind, die bereits in einer projektspezifischen Layoutbibliothek existieren, dann muss **LOGLIB** auch auf die entsprechende Projektdatei angewendet werden damit z.B. die richtige Pin/Gate-Swap-Information in die projektspezifische Bibliothek eingetragen wird. Bei falscher oder fehlender Anwendung von **LOGLIB** kommt es spätestens beim **Packager**-Lauf zu Fehlermeldungen wie etwa **`Bauteil nicht in Bibliothek!`**, **`Bauteil nicht definiert!`** oder **`Pin nicht gefunden!`**.

7.12 NETCONV

Name

netconv - Logical Netlist Conversion Utility

Synopsis

```
netconv projectname
```

Beschreibung

Das Programm **NETCONV** dient dazu, logische (d.h. ungepackte) ASCII-Netzlisten in den **Bartels AutoEngineer** zu übertragen.

NETCONV erwartet den Namen **projectname** des Projektes, für das die logische Netzliste umgesetzt werden soll. **NETCONV** liest die ASCII-Datei `<projectname>.net` und legt die darin enthaltene logische Netzliste unter dem Namen `netlist` in einem DDB-File mit Namen `<projectname>.ddb` ab. Diese logische Netzliste lässt sich mit dem **Packager** in eine physikalische (d.h. gepackte) Netzliste umwandeln. Anschließend kann das Layout erstellt werden, wobei auch Pin-/Gattertausch entsprechend den durch den **Packager** übertragenen `swap`-Kommandos (siehe hierzu auch Programm **LOGLIB**) durchgeführt werden kann.

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Die Eingabedatei muss mit dem Kommando `NETLIST;` beginnen und mit dem Kommando `END.` enden. Kommentare können an beliebiger Stelle zwischen `/*` und `*/` eingefügt werden.

Bauteilliste

Die Bauteilliste muss mit dem Schlüsselwort `PARTS` beginnen und kann für jedes Bauteil ein Kommando der Form

```
<part> : <llname> ;
```

enthalten. <part> ist der Bauteilname, und <llname> ist der Logical Library Name bzw. der Name des Stromlaufsymbols.

Netzliste

Die Netzliste muss mit dem Schlüsselwort `CONNECT` beginnen und für jedes Netz ein Kommando der Form

```
<part>.<pin>=<part>.<pin>=...=<part>.<pin>;
```

bzw.

```
/<net>/ <part>.<pin>=<part>.<pin>=...=<part>.<pin>;
```

enthalten. <net> ist hierbei der Netzname, <part> der Bauteilname und <pin> die Pinbezeichnung.

Beispiele

Inhalt der Datei **design.net**:

```
NETLIST;

/* Part list */

PARTS

        ic1 : 74ls00;

        ic2 : 74ls00;

        c1 : c;

/* Net list */

CONNECT

        ic1.a = ic2.y;

        ic2.a = ic1.y;

        /vcc/ c1.1;

        /vss/ c1.2;
END.
```

Übertragen der logischen Netzliste aus **design.net** in den **Bartels AutoEngineer** mit Hilfe von **NETCONV**:

```
>  netconv design ↵
```

Nach fehlerfreiem **NETCONV**-Lauf wurde die logische Netzliste aus **design.net** unter dem Namen **netlist** im DDB-File **design.ddb** abgespeichert. Diese logische Netzliste lässt sich nun mit Hilfe des **Packager** in eine physikalische umwandeln, und anschließend kann das entsprechende Layout erstellt werden.

Siehe auch

CONCONV, **LOGLIB**, **Packager**.

Diagnose

Die durch **NETCONV** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

Mit **NETCONV** können logische Netzlisten in den AutoEngineer übernommen werden. Sofern bei der Erstellung des zugehörigen Layouts Pin- und/oder Gattertausch vorgenommen wird, ist dies mit Hilfe der Backannotation in den Stromlauf zurückzumelden. Hierzu ist möglicherweise mit geeigneten Werkzeugen eine Auswertung des durch die Backannotation erzeugten Assignmentsdatei vorzunehmen.

In den Eingabedaten sind Bauteilnamen, Pinnamen oder Netznamen mit Sonderzeichen (**-**, **+**, **/**, **(**, **=**, usw.) in einfachen oder doppelten Anführungszeichen anzugeben.

7.13 REDASC

Name

redasc - REDAC ASCII Input Interface

Synopsis

```
redasc projectname [libraryfile]
```

Beschreibung

Das Programm **REDASC** erlaubt die Übernahme von Layoutdaten des Redac-MAXI-Systems, d.h. sowohl die Transformation von Layoutsymbolen als auch die Übertragung der Bauteilliste, der Netzliste und der Platzierung im CDI-Format in den **Bartels AutoEngineer**.

REDASC erwartet als Argument den Namen **`projectname`** der umzusetzenden CDI-Datei (diese Datei muss mit der Extension **`.cdi`** verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben).

REDASC erwartet optional als Argument den Namen **`libraryfile`** einer Layoutbibliothek (diese Datei muss im DDB(Design DataBase)-Format mit der Extension **`.ddb`** verfügbar sein; beim Programmaufruf ist diese Extension nicht mit anzugeben). Für den Fall, dass in der CDI-Datei nicht alle benötigten Layoutsymbole definiert sind, kann **REDASC** diese aus dem **libraryfile** laden (sofern natürlich darin vorhanden).

REDASC liest die CDI-Datei **`<projectname>.cdi`** und erzeugt das Design-File **`<projectname>.ddb`** mit den in der CDI-Datei definierten Layoutsymbolen, überträgt die in der CDI-Datei enthaltene Bauteilliste, Netzliste und Platzierung in das Design-File und erzeugt die Freelist **`<projectname>.fre`**. Die Freelist wird vom System nicht weiter benötigt und ist zur Auswertung durch den Benutzer bestimmt (Anzeige nicht angeschlossener Pins, Statistik). Das Design-File enthält nach dem **REDASC**-Lauf die benötigte (jobspezifische) Layoutbibliothek, die übersetzte Bauteil- und Netzliste sowie die Platzierung und kann im **Layouteditor** weiterbearbeitet werden.

Siehe auch

CONCONV

Diagnose

Die durch **REDASC** erzeugten Fehlermeldungen sind selbsterklärend.

7.14 RULECOMP

Name

rulecomp - Bartels Rule System Compiler

Synopsis

```
rulecomp srcfile [-l]
```

Beschreibung

Der Compiler **RULECOMP** dient der Übersetzung von Quellcodedateien zur Spezifikation von Regeldefinitionen für das im **Bartels AutoEngineer** integrierte **Neuronale Regelsystem**.

RULECOMP erwartet als erstes Argument den Quellcodedateinamen `srcfile` der Regeldefinition. (diese Datei muss mit der Extension `.rul` verfügbar sein; beim Programmaufruf ist der Dateiname ohne diese Extension anzugeben). **RULECOMP** übersetzt die angegebene Quellcodedatei und speichert die darin definierten Regeln bzw. Regelsätze in der Datei `brules.vdb` im BAE-Programmverzeichnis. Diese Regeln können später dann von speziellen BAE-Systemfunktionen bzw. von anwenderspezifischen **User Language**-Programmen angewendet bzw. abgearbeitet werden.

Optionen

Die Kommandozeilenoptionen des Regelcompilers bestehen aus einem Bindestrich (`-`) gefolgt von der Optionsspezifikation. Optionspezifikationen, die nur aus einem Buchstaben und der wahlweisen Angabe einer numerischen Modus- oder Schalterangabe bestehen, bezeichnet man häufig als Switches oder Flags.

Listing Option [-l]

Mit der Option `-l` kann die Listingausgabe gesteuert werden. Wenn die Option `-l` nicht angegeben ist, dann erfolgt keine Listingausgabe. Ist die Option `-l` angegeben, dann erzeugt der Rule System Compiler eine Listingausgabe für die kompilierten Regeln bzw. Regelsätze. Der Name der Listingdatei wird aus dem Namen der Quelltextdatei durch Abändern der Dateinamenserweiterung in `.lst` erzeugt. Die Listingdatei wird vom System nicht weiter benötigt, sie ist lediglich zur Auswertung durch den Benutzer bestimmt.

Beispiele

Kompilieren der in `routstd.rul` enthaltenen Regeldefinitionen; die übersetzten Regeldefinitionen werden unter dem Namen `routstd` in der Datei `brules.vdb` im BAE-Programmverzeichnis abgelegt:

```
> rulecomp routstd ↵
```

Kompilieren der in `routstd.rul` enthaltenen Regeldefinitionen mit Ausgabe eines Listings auf die Datei `routstd.lst`; die übersetzten Regeldefinitionen werden unter dem Namen `routstd` in der Datei `brules.vdb` im BAE-Programmverzeichnis abgelegt:

```
> rulecomp rulestd -l ↵
```

Dateien

`brules.vdb` -- BAE-Regeldatendatei (im BAE-Programmverzeichnis)

Siehe auch

Bartels AutoEngineer Benutzerhandbuch - Kapitel 6

Diagnose

Die durch **RULECOMP** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

RULECOMP ist ein mächtiges Programmierwerkzeug zur Erzeugung von Regeln zur automatisierten Generierung bzw. Bearbeitung von Designdaten im **Bartels AutoEngineer**. Wir empfehlen daher nachdrücklich, jede mit **RULECOMP** erzeugte Regel vor einem produktiven Einsatz zunächst in einer unkritischen Umgebung (Test-Software-Installation, Test-Job, Arbeitsdateien vorher sichern, ...) einem sorgfältigen Test zu unterziehen.

7.15 ULC - User Language Compiler

Name

ulc - Bartels User Language Compiler

Synopsis

```
ulc [-wcon|-wcoff] [[-S[ource]] srcfile...]

    [-lib libname...] [-dll libname...]

    [{-cp|-cl} [dstname...]]

    [-I[nclude] includepath...] [-D[efine] macroid...]

    [-O[0|1]] [-e[0|1]] [-w[0|1|2|3|4]] [-t[0|1]]

    [-l[0|1|2|3|4|5]] [-ld listingdirectory name]

    [-dp prgname...] [-dl libname...]

    [-ulp prgfilename] [-ull libfilename]

    [-log logfilename]
```

Beschreibung

Der **User Language Compiler** ulc dient dazu, **User Language**-Quelltext in **User Language**-Maschinenprogramme bzw. in **User Language**-Libraries zu übersetzen. **User Language**-Programme können vom **User Language Interpreter** ausgeführt werden. **User Language**-Libraries werden üblicherweise aus häufig benötigten Quelltexten erzeugt. Der Maschinencode von **User Language**-Libraries kann wahlweise statisch (während der Kompilierung durch den **User Language Compiler**) oder dynamisch (während der Laufzeit durch den **User Language Interpreter**) eingebunden werden in anderen Maschinencode (Programme oder Libraries). Der Vorteil des Librarykonzepts besteht darin, dass häufig benötigte Quelltexte *nur einmal* der zeitaufwändigen Kompilierung unterzogen werden müssen; anschließend kann der entsprechende Maschinencode über die sehr viel schnelleren Linkprozesse referenziert werden.

Optionen

Die Kommandozeilenoptionen des **User Language Compilers** bestehen aus einem Bindestrich (`-`) oder einem Schrägstrich (`/`) gefolgt von der Optionsspezifikation. Optionspezifikationen, die nur aus einem Buchstaben und der wahlweisen Angabe einer numerischen Modus- oder Schalterangabe bestehen, bezeichnet man häufig als Switches oder Flags. Diese speziellen Optionen können wahlweise gruppiert werden wie z.B. in **`/l2Ow3`** oder **`-O1w3l2`**, wo jeweils der Modus 2 für die Listingausgabe selektiert, der Optimierer aktiviert und der Warning Severity Level auf 3 gesetzt werden.

Wildcard Option [-wcon|-wcoff]

Mit Hilfe dieser Option kann die Berücksichtigung von Wildcards bei der Spezifikation von Datei- und Elementnamen aktiviert (Option **`-wcon`**; Default) bzw. deaktiviert (Option **`-wcoff`**) werden. Ist die Wildcarderkennung aktiviert, dann erlangen die Zeichen **?** und ***** Sonderbedeutung bei der Spezifikation von Datei- und Elementnamen; **?** ist dann Platzhalter für ein beliebiges Zeichen, ***** ist dann Platzhalter für eine beliebige Anzahl beliebiger Zeichen. Die Deaktivierung der Wildcarderkennung ist notwendig, um die Bearbeitung von Programmnamen wie z.B. **SCM_?** oder **GED_*** zu ermöglichen.

Source File Option [[-S[ource]] srcfile...]

Mit dieser Option werden die Namen der zu übersetzenden Quelltextdateien spezifiziert. Die Dateinamen dürfen dabei Verzeichnispfade enthalten, d.h. die Quelltextdateien müssen nicht notgedrungen im aktuellen Verzeichnis abgelegt sein. Bei der Auswertung von Dateinamen werden Wildcards berücksichtigt, sofern die Wildcarderkennung mit der Option **`-wcon`** (siehe oben) aktiviert ist. Dateinamen können wahlweise mit oder ohne Namenserweiterung spezifiziert werden. Wird die Namenserweiterung weggelassen, dann für der Compiler automatisch die Extension **`.ulc`** an die entsprechenden Dateinamen an. Quelltextdateinamen mit anderen Namenserweiterungen der müssen also mit ihrer Extension spezifiziert werden. Damit ist es dann allerdings auch möglich, z.B. **User Language**-Libraries aus Includedateien mit der Extension **`.ulh`** zu erzeugen. Der Type des zu erzeugenden Maschinencodes wird mit den Optionen **`-cp`** (für **User Language**-Programme; siehe unten) bzw. **`-cl`** (für **User Language**-Libraries) festgelegt. Der Elementname des erzeugten Maschinencodes ergibt sich aus dem Quelltextdateinamen, wobei der Verzeichnispfad und die Dateinamenserweiterung weggelassen werden. Ein von dieser Konvention abweichender Name für den Maschinencode kann mit den Optionen **`-cp`** bzw. **`-cl`** (siehe unten) spezifiziert werden. Das Schlüsselwort **`-Source`** bzw. **`-S`** ist bei der Spezifikation von Quelltextdateinamen nur dann erforderlich, wenn sich sonst Mehrdeutigkeiten bei der Auswertung der Kommandozeile ergeben könnten. Dies ist z.B. dann nicht der Fall, wenn die Quelltextdateinamen die ersten Namensargumente im Compileraufruf darstellen. Mit Hilfe der Schlüsselworte **`-Source`** bzw. **`-S`** können andererseits jedoch an beliebigen Stellen in der Kommandozeile Quelltextdateien spezifiziert. Ist weder die Option **`-dp`** noch die Option **`-dl`** (siehe unten) spezifiziert, dann erwartet der **User Language Compiler** die Angabe von zumindest einer Quelltextdatei.

Static Link Option [-lib libname...]

Die Static Link Option **`-lib`** erwartet den Namen einer oder mehrerer **User Language**-Libraries sowie die Spezifikation von zumindest einer Quellcodedatei (siehe oben, Option **`-Source`**). Die mit **`-lib`** angeforderten Libraries müssen in kompilierter Form in der Datei **`ulcprog.vdb`** im BAE-Programmverzeichnis verfügbar sein. Der im **User Language Compiler** integrierte Linker bindet den Maschinencode der angeforderten Libraries in den Maschinencode der zu übersetzenden Quelltextdateien ein.

Dynamic Link Option [-dll libname...]

Die Dynamic Link Option **`-lib`** erwartet den Namen einer oder mehrerer **User Language**-Libraries sowie die Spezifikation von zumindest einer Quellcodedatei (siehe oben, Option **`-Source`**). Die mit **`-dll`** angeforderten Libraries müssen in kompilierter Form in der Datei **`ulcprog.vdb`** im BAE-Programmverzeichnis verfügbar sein. Der im **User Language Compiler** integrierte Linker erzeugt die für den **User Language Interpreter** notwendigen Informationen zur Laufzeiteinbindung der angeforderten Libraries in den Maschinencode der zu übersetzenden Quelltextdateien.

Create Program/Library Option [{-cp|-cl} [dstname...]]

Mit dieser Option wird der Typ des zu erzeugenden Maschinencodes festgelegt. Der **User Language Compiler** kann sowohl **User Language**-Programme als auch **User Language**-Libraries generieren. Wenn weder die Option **`-cp`** noch die Option **`-cl`** angegeben ist, dann erzeugt der Compiler **User Language**-Programme. Mit der Option **`-cp`** kann die Generierung von Programmen explizit veranlasst werden, die Option **`-cl`** hingegen veranlasst die Generierung von Libraries; es dürfen nicht beide Optionen gleichzeitig angegeben werden. Der Elementname des zu erzeugenden Maschinencodes ergibt sich standardmäßig aus dem jeweiligen Quellcodedateinamen durch Elimination des Verzeichnispfades und der Namenserweiterung. Abweichend von dieser Konvention können mit den Optionen **`-cp`** und **`-cl`** explizit andere Elementnamen für zu erzeugenden Maschinencode angegeben werden, wobei dann aber nur noch *genau eine* Quelltextdatei spezifiziert werden darf (siehe oben, Option **`-Source`**). Der generierte Maschinencode wird unter dem spezifizierten Zielelementnamen in der Datei **`ulcprog.vdb`** im BAE-Programmverzeichnis abgelegt. Wildcards werden bei der Angabe von Zielelementnamen für Maschinencode grundsätzlich nicht berücksichtigt. Die explizite Angabe mehrerer Zielelementnamen dagegen ist zulässig; damit ist es möglich den Maschinencode einer einzelnen Quelltextdatei unter verschiedenen Namen abzulegen, also z.B. aus der Quelltextdatei **bae_st.ulh** mit einem einzigen Compileraufruf die Programme **SCM_ST**, **GED_ST**, usw. zu erzeugen.

Include Path Option [-I[nclude] includepath...]

Mit der Option **`-Include`** (bzw. **`-I`**) können Alternativpfade für die Suche nach Includedateien spezifiziert werden. Diese Option erwartet zumindest einen Verzeichnispfadnamen als Argument. Stößt der Compiler im Quelltext auf eine **`#include`**-Anweisung, dann sucht er zunächst im aktuellen Verzeichnis nach der angeforderten Includedatei und dehnt die Suche anschließend auf die mit der Option **`-Include`** angegebenen Verzeichnisse aus, wobei die Verzeichnisse in der Reihenfolge ihrer Spezifikation abgesucht werden.

Define Option [-D[efine] macroid...]

Mit der Option **-Define** (bzw. **-D**) können beim Compileraufruf Makros definiert werden. Diese Option erwartet zumindest einen Makronamen als Argument. Die Option **-Define** entspricht der **#define**-Anweisung im Quelltext, d.h. die mit dieser option definierten Makros können in den Preprozessoranweisungen **#ifdef** und **#ifndef** zur Steuerung der bedingten Übersetzung ausgewertet werden.

Optimizer Option [-O[0|1]]

Mit der Option **-O** kann der Optimierer des **User Language Compilers** aktiviert bzw. deaktiviert werden. Per Default (d.h., wenn diese Option nicht spezifiziert ist) ist der Optimierer deaktiviert. Die Option **-O** bzw. **-O1** aktiviert den Optimierer. Mit der Option **-O0** kann der Optimierer explizit deaktiviert werden. Der Optimierer befreit den Maschinencode von Redundanzen, und gestaltet ihn durch Modifikationen effizienter. Optimierter Maschinencode benötigt in der Regel erheblich weniger Festplatten- und Hauptspeicher und kann schneller geladen und abgearbeitet werden. Es wird daher dringend empfohlen, den Optimierer zu aktivieren.

Error Severity Option [-e[0|1]]

Mit der Option **-e** kann der Error Severity Level gesetzt werden. Per default (d.h., wenn diese Option nicht spezifiziert ist) ist der Wert 1 eingestellt. Die Option **-e0** setzt den Error Severity Level auf 0; die Option **-e** bzw. **-e1** setzt den Error Severity Level explizit auf 1. Ist der Error Severity Level auf 1 eingestellt, dann versucht der Compiler *alle* beim Compileraufruf spezifizierten Quelltextdateien zu übersetzen (ungeachtet etwaiger Fehler beim übersetzen einzelner Quelltexte); ein Error Severity Level von 0 hingegen veranlasst den Compiler, den Übersetzungsvorgang bei der ersten fehlerhaften Quelltextdatei abzubrechen.

Warning Severity Option [-w[0|1|2|3|4]]

Mit der Option **-w** kann der Warning Severity Level auf einen Wert von 0 bis 4 gesetzt werden. Per Default (d.h., wenn diese Option nicht spezifiziert ist) ist der Wert 0 eingestellt. Wird diese Option ohne die explizite Angabe eines Levels angegeben (**-w**), dann wird der Wert 3 eingestellt. Jede im Compiler definierte Warnmeldung ist einem speziellen Warning Severity Level zugeordnet, wobei höhere Werte unwichtigere Warnungen kennzeichnen. Der Compiler gibt nur die Warnungen aus, deren Warning Severity Level kleiner oder gleich dem mit der Option **-w** eingestellten Level ist, d.h. mit dieser Option kann die Ausgabe weniger bedeutsamer Warnmeldungen unterdrückt werden.

Top Level Warnings Only Option [-t[0|1]]

Mit der Option **-t** können wahlweise Warnungen mit Bezug auf die Kompilierung von Includedateien unterdrückt werden (Wert 1). Per Default (d.h., wenn diese Option nicht spezifiziert ist bzw. der Optionswert 0 gesetzt ist) werden Warnmeldungen mit Bezug auf alle kompilierten Quellcodedateien ausgegeben. Ist dieser Optionswert auf 1 gesetzt, dann unterdrückt der **User Language Compiler** die Ausgabe von Warnmeldungen mit Bezug auf die Kompilierung von Includedateien und gibt lediglich die Warnmeldungen mit Bezug auf die "Top-Level"-Quellcodedatei(en) aus. Dies reduziert die Anzahl der Warnmeldungen und erleichtert damit deren Analyse insbesondere dann, wenn mit Standardincludedateien gearbeitet wird, die (eine Vielzahl von) Funktionen und Variablen enthalten, welche nicht in jedem Programm verwendet werden.

Listing Option [-l[0|1|2|3|4|5]]

Mit der Option **-l** kann die Listingausgabe gesteuert werden. Dabei können Listingmode im Bereich von 0 bis 5 spezifiziert werden. Mit dem Listingmodus 0 wird keine Listingausgabe erzeugt, während der Modus 5 die detailliertesten Informationen liefert. Modus 0 ist der Standardwert, d.h. wenn die Option **-l** nicht angegeben ist, dann erfolgt auch keine Listingausgabe. Wird diese Option ohne die explizite Angabe des Modus spezifiziert (**-l**), dann wird der Modus 5 eingestellt. Der Name der Listingdatei wird aus dem Namen der Quelltextdatei durch Abändern der Dateinamenserweiterung in **.lst** erzeugt. Die Listingdatei wird vom System nicht weiter benötigt, sie ist lediglich zur Auswertung durch den Benutzer bestimmt.

Listing Directory [-ld listingdirectoryname]

Die Option **-ld** gestattet die Spezifikation eines alternativen Verziechnisses für die mit der Option **-l** auszugebenden Listindateien. Dies ist insbesondere bei der Verwendung von **make**-Utilities zur automatisierten Compilierung geänderter **User Language**-Programme nützlich, um das Quellverzeichnis "sauber" zu halten. Im Verzeichnis **baeulc** wird ein **makefile** bereitgestellt, das die Abhägigkeiten der **User Language**-Programme von Includedateien enthält und mit Listingdateien in einem Unterverzeichnis (**lst**) arbeitet.

Delete Program Option [-dp prgname...]

Mit der Option **-dp** können einmal übersetzte **User Language**-Programme wieder aus der Datei **ulcprog.vdb** im BAE-Programmverzeichnis gelöscht werden. Diese Option erwartet den Elementnamen zumindest eines Programms als Argument. Bei der Auswertung der Elementnamen werden Wildcards berücksichtigt, sofern die Wildcarderkennung mit der Option **-wcon** (siehe oben) aktiviert ist. Beim Versuch, nicht-existente Libraries zu löschen, werden Warnungen ausgegeben. Damit nicht Maschinencode gelöscht werden kann, der unmittelbar zuvor im gleichen Compilerlauf erzeugt wurde, erfolgt das Löschen von Programmen grundsätzlich immer *bevor* mit der Übersetzung von Quelltexten begonnen wird.

Delete Library Option [-dl libname...]

Mit der Option **-dl** können einmal übersetzte **User Language**-Libraries wieder aus der Datei **ulcprog.vdb** im BAE-Programmverzeichnis gelöscht werden. Diese Option erwartet den Elementnamen zumindest einer Library als Argument. Bei der Auswertung der Elementnamen werden Wildcards berücksichtigt, sofern die Wildcarderkennung mit der Option **-wcon** (siehe oben) aktiviert ist. Beim Versuch, nicht-existente Libraries zu löschen, werden Warnungen ausgegeben. Damit nicht Maschinencode gelöscht werden kann, der unmittelbar zuvor im gleichen Compilerlauf erzeugt wurde, erfolgt das Löschen von Libraries grundsätzlich immer *bevor* mit der Übersetzung von Quelltexten begonnen wird.

Program Database File Name Option [-ulp prgfilename]

Per Default speichert der **User Language Compiler User Language**-Programme in der Datei **ulcprog.vdb** im Programmverzeichnis des **Bartels AutoEngineer** ab. Mit der Option **-ulp** kann hierfür eine andere Datei angegeben werden.

Library Database File Name Option [-ull libfilename]

Per Default speichert der **User Language Compiler User Language**-Libraries in der Datei **ulcprog.vdb** im Programmverzeichnis des **Bartels AutoEngineer** ab. Mit der Option **-ull** kann hierfür eine andere Datei angegeben werden.

Log File Option [-log logfilename]

Der **User Language Compiler** gibt alle Meldungen auf die Standardausgabe und gleichzeitig auf eine Reportdatei aus. Die Ausgabe auf die Reportdatei erfolgt, um längere Listen von Meldungen, welche insbesondere bei der Übersetzung mehrerer Quelltextdateien entstehen können, zu sichern. Der Standardname der Reportdatei ist **ulc.log** (im aktuellen Verzeichnis). Mit der option **-log** kann ein anderer Dateiname für die Reportdatei angegeben werden.

Beispiele

Kompilieren des in **ulprog.ulc** enthaltenen **User Language**-Programms mit Optimierung und Ausgabe von Warnmeldungen; das übersetzte Programm wird unter dem Namen **ulprog** in der Datei **ulcprog.vdb** im BAE-Programmverzeichnis abgelegt:

```
> ulc ulprog -Ow ↵
```

Kompilieren des in **ulprog.ulc** enthaltenen **User Language**-Programms mit Erzeugung einer Listingdatei (**ulprog.lst**); das übersetzte Programm wird unter dem Namen **newprog** in der Datei **ulcprog.vdb** im BAE-Programmverzeichnis abgelegt:

```
> ulc ulprog -l -cp newprog ↵
```

Löschen der **User Language**-Programme mit Namen **ulprog** und **newprog** sowie der **User Language**-Libraries, deren Namen mit **test** beginnen und auf **lib** enden, aus der Datei **ulcprog.vdb** im BAE-Programmverzeichnis:

```
> ulc -dp ulprog newprog -dl test*lib ↵
```

Erzeugen der **User Language**-Library **libsll** aus der Quelltextdatei **libbae.ulh** (der Optimierer ist aktiviert; ein Listing wird auf die Datei **libbae.lst** ausgegeben):

```
> ulc libbae.ulh -cl libsll -l2O ↵
```

Kompilieren aller im aktuellen Verzeichnis enthaltenen Quelltextdateien mit der Extension **.ulc** sowie statisches Linken der generierten Maschinenprogramme mit der **User Language**-Library **libsll** (das Makro **USELIB** wird zur Kontrolle der bedingten Übersetzung definiert; der Optimierer ist aktiviert):

```
> ulc *.ulc -Define USELIB -lib libsll -O ↵
```

Generieren der **User Language**-Libraries **libstd** und **stdlib** aus der Quelltextdatei **std.ulh** (der Optimierer ist aktiviert, Warnmeldungen mit Severity Level kleiner oder gleich 2 werden ausgegeben):

```
> ulc -w2 -O -cl libstd stdlib -Source std.ulh ↵
```

Generieren der **User Language**-Library **liblay** aus der Quelltextdatei **\baeulc\lay.ulh** mit dynamischen Linken der Library **libstd** (der Optimierer ist aktiviert, der Warning Severity Level ist auf den Standardwert 3 gesetzt, die Compiler-Meldungen werden auf die Reportdatei **genlib.rep** anstelle **ulc.log** ausgegeben):

```
> ulc /wO -cl liblay -S \baeulc\lay.ulh -dll libstd -log genlib.rep ↵
```

Generieren der **User Language**-Programm **laypcr** und **tracerep** aus den Quelltextdateien **laypcr.old** und **tracerep.ulc** mit dynamischem Linken der Library **liblay** (der Optimierer ist aktiviert):

```
> ulc laypcr.old /dll liblay /cp -O /S tracerep ↵
```

Dateien

`ulcprog.vdb` -- BAE-**User Language**-Datenbank (im BAE-Programmverzeichnis)

Siehe auch

USERLIST, **User Language Interpreter**, Bartels User Language Programmierhandbuch

Diagnosis

Die durch **ULC** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

ULC ist ein Programmierwerkzeug zur Erzeugung von Programmen, die die Manipulation von DDB-Dateiinhalten ebenso wie die Erzeugung von Fertigungsdaten zulassen. Auch eine weit reichende Änderung der Benutzeroberfläche ist durch die mit **ULC** erzeugten Programme möglich. Daher sollte jedes mit **ULC** erzeugte Programm vor einem produktiven Einsatz in jedem Fall zunächst in einer unkritischen Umgebung (Test-Software-Installation, Test-Job, Arbeitsdateien vorher sichern, ...) einem sorgfältigen Test unterzogen werden. In diesem Zusammenhang muss auch eindringlich auf die Möglichkeit des Missbrauchs durch Dritte (Einspielen zerstörerisch wirkender Programme in Ihr eigenes `ulcprog.vdb`) hingewiesen werden.

7.16 User Language Interpreter

Beschreibung

Der **Bartels User Language Interpreter** erlaubt die Ausführung von kompilierten, d.h. durch den **Bartels User Language Compiler** generierten **Bartels User Language**-Programmen. Der **Bartels User Language Interpreter** ist in nahezu allen Programmmodulen des **Bartels AutoEngineer** eingebunden.

Aufruf von User Language-Programmen

In dem Menü Datei ist im **Schaltplaneditor**, im **Layouteditor**, im **Neuronalen Autorouter**, im **CAM-Prozessor**, im **CAM-View**-Modul und im **Chipeditor** des **Bartels AutoEngineer** ein Menüpunkt Anwenderfunktion vorhanden, mit dem es möglich ist, ein in **ulcprog.vdb** (im BAE-Programmverzeichnis) gespeichertes **User Language**-Programm über seinen Namen zu starten.

Der Start eines **User Language**-Programms ist auch durch Betätigung einer Standardtaste (0, 1, ..., 9, a, b, c, ...) oder einer Funktionstaste (F1, F2, ..., F12) möglich, sofern sich der Anwender in der Menüleiste befindet; hierzu muss in **ulcprog.vdb** ein **User Language**-Programm mit dem Namen **scm_<c>** bzw. **scm_f<n>** (im **Schaltplaneditor**), **ged_<c>** bzw. **ged_f<n>** (im **Layouteditor**), **ar_<c>** bzw. **ar_f<n>** (im **Autorouter**), **cam_<c>** bzw. **cam_f<n>** (im **CAM-Prozessor**), **cv_<c>** bzw. **cv_f<n>** (im **CAM-View**-Modul) oder **ced_<c>** bzw. **ced_f<n>** (im **Chipeditor**) verfügbar sein. **<c>** gibt dabei das gedrückte Zeichen, **f<n>** die gedrückte Funktionstaste an.

Sind Programme mit den Namen **scm_st**, **ged_st**, **ar_st**, **cam_st**, **cv_st** bzw. **ced_st** vorhanden, so werden diese automatisch beim Start des entsprechenden Programmoduls ausgeführt.

Über die **Bartels User Language** werden darüber hinaus Systemfunktionen zur Tastaturprogrammierung und Menübelegung zur Verfügung gestellt. Durch eine geeignete Verwendung dieser Funktionen (z.B. in den **User Language**-Startupprogrammen) lässt sich die Benutzeroberfläche des **Bartels AutoEngineer** sogar während der Bearbeitung dynamisch verändern. Und schließlich besteht über eine weitere **User Language**-Systemfunktion die Möglichkeit **User Language**-Programme aus anderen **User Language**-Programmen heraus aufzurufen.

Beispiele

Aufruf des **User Language**-Programms **ulprog**:



Dateien

ulcprog.vdb -- BAE-**User Language**-Datenbank im BAE-Programmverzeichnis

Siehe auch

User Language Compiler, Schematic Editor, Layouteditor, Autorouter, CAM-Prozessor, CAM-View, **USERLIST**, Bartels User Language - Programmierhandbuch

Diagnose

Die durch den **User Language Interpreter** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

Der **Bartels User Language Interpreter** ist ein Programm, das die Manipulation von DDB-Dateiinhalten ebenso wie die Erzeugung von Fertigungsdaten zulässt. Auch eine weit reichende Änderung der Benutzeroberfläche ist durch den Einsatz des **Bartels User Language Interpreters** mmöglich. Daher sollte jedes **User Language**-Programm vor einem produktiven Einsatz in jedem Fall zunächst in einer unkritischen Umgebung (Test-Software-Installation, Test-Job, Arbeitsdateien vorher sichern, ...) einem sorgfältigen Test unterzogen werden. In diesem Zusammenhang muss auch eindringlich auf die Möglichkeit des Missbrauchs durch Dritte (Einspielen zerstörerisch wirkender Programme in Ihr eigenes **`ulcprog.vdb`**) hingewiesen werden.

7.17 USERLIST

Name

userlist - User Programmable List Generator

Synopsis

```
userlist scriptfile projectfile jobname
```

Beschreibung

Das Programm **USERLIST** ist ein anwenderprogrammierbarer ASCII-Listengenerator, mit dem Daten aus Design-File-Netzlisten in einem vom Benutzer frei festlegbaren Format extrahiert werden können. Dabei lassen sich die Art der Ausgabe (Vor-/Nachkommastellen, Ausgabe von Schlüsselwörtern, usw.) und der Typ der auszugebenden Information (Bauteilattribute, Pinattribute, usw.) vom Benutzer in einem Spezifikationsfile (userlist-Script) frei definieren.

Das Programm **USERLIST** interpretiert das vom Anwender erstellte **USERLIST**-Script **<scriptfile>.usf**, liest die unter **<jobname>** im Design-File **<projectfile>.ddb** abgelegte Netzliste und generiert eine Ausgabedatei mit Namen **<projectfile>.<ext>** (die Extension **<ext>** ist im Script zu definieren).

Format der Eingabedatei

Dateianfang, Dateiende, Kommentare

Am Dateianfang des **USERLIST**-Scripts ist die Extension des Ausgabefiles über den Parameter **EXTENSION** zu definieren:

```
EXTENSION = "<ext>";
```

Das **USERLIST**-Script muss mit dem Schlüsselwort **ENDSPEC** enden. Kommentare können zwischen **/*** und ***/** eingefügt werden.

FOR-Kommando

Die Auswahl von Elementen eines Bereiches erfolgt mit

```
FOR (ALL <class> ) { <commands> }
```

<class> gibt die Klasse der zu bearbeitenden Objekte an. Hierfür können die Schlüsselworte **NETS**, **PARTS**, **PINS**, **ATTRIBUTES** bzw. **<attname>** eingesetzt werden. **NETS** bezeichnet die Objektklasse Netzliste, **PARTS** die Bauteilliste, **PINS** die Pinliste und **ATTRIBUTES** die Attributliste. **<attname>** erlaubt die Abarbeitung der Attributwertliste für das Attribut mit dem durch **<attname>** gegebenen Namen. Die Kommandoliste **<commands>** wird für jedes Element der angegebenen Objektklasse einmal abgearbeitet. Dabei können, soweit sinnvoll, weitere **FOR**-Schleifen verschachtelt werden. Bei verschachtelten **FOR**-Schleifen beziehen sich die Objektklassen jeweils auf die in der übergeordneten Schleife angegebene Objektklasse. So arbeitet in

```
FOR (ALL NETS) { FOR (ALL PINS) {...}}
```

die **PINS**-Schleife jeweils die Pins des aktuellen Netzes der **NETS**-Schleife ab.

Ausgabe-Befehle

Zur Ausgabe stehen die Kommandos **PRINT** und **PRINTFOR** zur Verfügung. Die Syntax für das Kommando **PRINT** lautet:

```
PRINT(<parameters>);
```

In **<parameters>** ist - getrennt durch Kommas - die Liste der auszugebenden Parameter enthalten. In Anführungszeichen angegebene Parameter werden unverändert ausgegeben. Anführungszeichen selbst werden mit dem Schlüsselwort **QUOTES** ausgegeben. Tabulatorzeichen können mit **TAB** erzeugt werden. Eine neue Zeile erreicht man durch Angabe von **CR**. Die restlichen Parameter geben Elemente der Datenbank an. Dabei kann durch Angabe von

```
:<length>:<decimals>
```

die Anzahl der Gesamtstellen und Nachkommastellen zur Ausgabe von Zahlen und Strings angegeben werden. Wird **<length>** negativ angegeben, so wird der Ausgabestring linksbündig in dem durch die Gesamtstellen definierten Ausgabefeld dargestellt. Mit

```
%<length>:<decimals>
```

werden Leerstellen zu Beginn des Ausgabefeldes mit Nullen aufgefüllt. Standardeinstellung sind 3 Nachkommastellen und eine Feldbreite, die der Länge des darzustellenden Elements entspricht. Reichen die angegebenen Gesamtstellen nicht zur Ausgabe aus, so verbreitert sich das Ausgabefeld entsprechend. Bei Elementen, die Distanzen angeben, kann die Ausgabe durch Nachstellen von **" MM"** bzw. **" INCH"** in der entsprechenden Längeneinheit erfolgen. Beispiele für die Anwendung des **PRINT**-Kommandos sind:

```
PRINT(QUOTES,PINWIDTH:7:3,QUOTES);              /* Output: "  3.756" */

PRINT(QUOTES,PINWIDTH%7:3,QUOTES);              /* Output: "003.756" */

PRINT(QUOTES,PINWIDTH:-7:3,QUOTES);             /* Output: "3.756  " */
```

In **PRINT**-Kommandos kann hinter der Angabe von Namen und Attributen mit durch Leerzeichen getrennten **UPPER**- oder. **LOWER**-Texten eine Ausgabe der Namen bzw. Attribute in Groß- bzw. Kleinschreibung erzwungen werden.

Das Kommando **PRINTFOR** dient zur Ausgabe von Listen unbekannter Länge mit Trennzeichen zwischen den einzelnen Elementen. Die Syntax für dieses Kommando lautet:

```
PRINTFOR (ALL <class>) SEPERATOR(<sep>),ELEMENTS(<elements>);
```

Die Trennzeichen zwischen den einzelnen Elementen werden über den Seperator **<sep>** spezifiziert (z.B. **.**, **CR**, usw.). In <elements> stehen die Daten, die zu einem Listenelement ausgegeben werden sollen. Für die Parameterliste in <sep> und <elements> gilt die gleiche Syntax, wie für die Parameterliste in **PRINT**. Steht **PRINTFOR** innerhalb einer **FOR**-Schleife, so beschränkt sich die Ausgabe automatisch auf das gerade aktuelle Element der **FOR**-Schleife.

IF-Abfragen

Das **IF**-Kommando dient zur bedingten Abarbeitung von Kommandos. Die Syntax hierfür lautet:

```
IF (<expr>) { <commands> }
```

bzw.

```
IF (<expr>) { <commands> } ELSE { <commands> }
```

Die Bedingungsanweisung **<expr>** besteht entweder aus

```
? <attr>
```

oder

```
<attr> <operator> <attr|constant>
```

Der Ausdruck **? <attr>** prüft, ob das mit **<attr>** angegebene Attribut vorhanden ist. Für vergleichende Ausdrücke stehen die Operatoren (**<operator>**) **=** (gleich), **<>** (ungleich), **<** (kleiner), **>** (größer), **<=** (kleiner oder gleich), **>=** (größer oder gleich) zur Verfügung.

Zählvariable

Die Befehle

```
CLEARCOUNTER;
```

und

```
COUNTUP;
```

dienen zur Ansteuerung eines Zählers. Mit **CLEARCOUNTER** wird der Zähler auf Null gesetzt. **COUNTUP** erhöht den Zähler um eins. Der aktuelle Wert des Zählers steht als Attribut **COUNTVALUE** zur Verfügung.

Attribute

Es können folgende Attribute ausgegeben werden:

Netzdaten	**NETNAME**	Netzname
	NETPINCOUNT	Anzahl der Pins am Netz
	PRIORITY	Netz-Priorität (für Router)
	MINDIST	Netz-Mindestabstand (für Router)
	NETNUMBER	Netznummer
Bauteildaten	**PARTNAME**	Bauteilname
	PINCOUNT	Anzahl der Pins am Bauteil
	FREEPINS	Anzahl der nichtangeschlossenen Pins am Bauteil
	PARTATTRIBCOUNT	Anzahl der Attribute des Bauteiles
	$attributname	Wert des angegebenen Attributes
Pindaten	**PINNAME**	Pinname
	PINWIDTH	Pinbezogene Anschlussbreite
Allgemeine Daten	**PROJECTNAME**	Projekt- bzw. Design-Name
	ATTRIBCOUNT	Gesamthäufigkeit der aktuellen Kombination Attributname, Attributwert
	ATTRIBNAME	Name des aktuellen Attributes
	ATTRIBVALUE	Wert des aktuellen Attributes
	COUNTVALUE	Aktueller Wert des Counters

Beispiele

Netzlisten-Generator **conconv.usf**:

```
/* Connection List Generator */

EXTENSION = ".con";

PRINT ("LAYOUT ", PROJECTNAME, ";", CR);

PRINT ("PARTS",CR);

FOR (ALL PARTS)

{

    PRINT ("  ",PARTNAME," : ",$plname,";",CR);

}

PRINT ("CONNECT",CR);

FOR (ALL NETS)

{

    PRINT ("  /", NETNAME,"/ ");

    PRINTFOR (ALL PINS)

        SEPERATOR ("="), ELEMENTS (PARTNAME,".",PINNAME);

    PRINT (";",CR);

}

PRINT ("END.",CR);

ENDSPEC
```

Durch einen Aufruf der Form

```
> userlist conconv design board ↵
```

wird die Netzliste **board** des Job-Files **design.ddb** gelesen und auf die Datei **design.con** ausgegeben. Die Ausgabe hat dann folgendes Format:

```
LAYOUT board;

PARTS

        ic1 : dil14;

        ic2 : dil14;

        ic3 : dil16;

CONNECT

        /gnd/ ic1.1=ic2.2=ic3.9;

        /vcc/ ic1.11=ic2.5=ic3.7;

END.
```

Partlister:

```
EXTENSION = ".ptl";

FOR (ALL $plname)

{

    PRINT (ATTRIBCOUNT," ",ATTRIBVALUE,CR);

}

ENDSPEC
```

Ausgabedatei **<job>.ptl**, z.B.:

```
3 cap50

4 dil14

2 dil16

1 r75
```

Diagnose

Die durch **USERLIST** erzeugten Fehlermeldungen sind selbsterklärend.

7.18 VALCONV

Name

valconv - VALID to Bartels Conversion

Synopsis

```
valconv projectname
```

Beschreibung

Das Programm **VALCONV** erlaubt die Übertragung von Bauteil- und Netzlisten, die mit dem VALID-Stromlaufeditor erstellt wurden, in den **Bartels AutoEngineer**.

Das Argument **projectname** gibt den Namen des DDB(Design DataBase)-Files an, in das die VALID-Bauteil- und die VALID-Netzliste zu transferieren sind (es wird eine Datei mit Namen `<projectname>.ddb` generiert; die Extension `.ddb` ist beim Programmaufruf nicht mit anzugeben).

VALCONV liest die VALID-Bauteilliste aus der Datei mit Namen `pstxprt`, die VALID-Netzliste wird aus der Datei `pstxnet` gelesen. Diese beiden Dateien werden vom VALID-System generiert und müssen sich beim **VALCONV**-Aufruf im aktuellen Verzeichnis befinden.

Nach erfolgreichem **VALCONV**-Lauf befindet sich im generierten DDB-File eine logische, d.h. eine ungepackte Netzliste. Diese Netzliste muss im **Bartels AutoEngineer** mit Hilfe des **Packagers** in eine physikalische, d.h. eine gepackte Netzliste umgesetzt werden. Erst dann kann im **Layouteditor** auf der Basis dieser Netzliste das Layout erstellt werden.

Dateien

`pstxprt` -- VALID-Bauteilliste
`pstxnet` -- VALID-Netzliste

Siehe auch

NETCONV, **Packager**, **LOGLIB**

Diagnose

Die durch **VALCONV** erzeugten Fehlermeldungen sind selbsterklärend.

Warnungen

In den Eingabedaten sind Bauteilnamen, Pinnamen oder Netznamen mit Sonderzeichen (`-`, `+`, `/`, `(`, `=`, usw.) in einfachen oder doppelten Anführungszeichen anzugeben.